ÉCOLE NATIONALE DES CHARTES

Paul, Hector Kervegan

Traitement, exploitation et analyse d'un corpus semi-structuré : le cas des catalogues de vente de manuscrits

Mémoire pour le diplôme de master

Technologies numériques appliquées à l'histoire

2022

Résumé

Introduction

J'ai choisi de structurer mon mémoire autour de plusieurs questions connexes, qui, à différents degrés, se retrouvent tout au long du développement :

En quoi la nature semi-structurée du corpus permet d'en automatiser le traitement ? Comment produire des informations normalisées et exploitables à partir d'un corpus textuel semi-structuré ? En quoi ce traitement et la traduction des documents vers d'autres formats et d'autres médias impacte leur réception ? Quels sont les choix techniques qui influencent cette réception ?

Les deux premières questions, d'orientation plutôt technique, forment la colonne vertébrale pour le mémoire ; elles lient deux aspects centraux : la nature du corpus et la manière dont sa structure permet toute la chaîne de traitement. Par « semi-structuré », j'entends que, à un niveau distant, toutes les entrées de catalogue suivent la même structure ; des séparateurs distinguent les différentes parties, et les informations sont souvent structurées de manière semblable pour chaque manuscrit vendu. Cela permet un traitement de « basse technologie » (*low-tech*) en évitant d'entraîner de lourds modèles de traitement du langage naturel (ce qui aboutirait à des solutions complexes, difficiles à maintenir et à faire évoluer et relativement opaques dans leur fonctionnement). À l'inverse, un corpus semi-structuré peut être traité en déduisant une « structure abstraite », que chaque entrée de catalogue partage. Il est alors possible de mettre en place des solutions techniques plus faciles, pour un résultat de qualité équivalente. Produire des « informations normalisées et exploitables » implique de traiter le corpus en cherchant des réponses à des questions de recherche précises – dans le cadre de mon stage, une question centrale a été de chercher à isoler les facteurs déterminant le prix d'un manuscrit.

Les deux dernières questions, au premier abord plus théoriques, me semblent centrales, notamment à la troisième partie de ce mémoire. Numérisation, traitement informatisé et diffusion sur le web ne sont pas des opérations neutres, mais un ensemble de « traductions » des documents originels. Ces processus comportent une part de choix conscients, qu'il s'agit de mettre en avant. Par exemple, on considère que la majorité des documents vendus ont pour titre l'auteur.ice du document. Cette personne n'est cependant pas toujours mentionnée, et des documents peuvent être nommés d'après un lieu, un évènement ou un thème (la Révolution française, par exemple). Ces « traductions » des catalogues sont relativement discrètes tout au long de la chaîne de traitement (où le

format dominant est la `TEI`, qui garde une relation d'équivalence avec le texte). C'est lors du passage au site web que ce processus de traduction devient plus évident, et, potentiellement, plus problématique. On y abandonne la référence au document originel (les catalogues numérisés ne sont pas accessibles en ligne par un.e utilisateur.ice), le catalogue n'est plus la manière privilégiée d'accéder aux items vendus... De plus, la construction d'un site web implique la conception d'une interface et, dans notre cas, la production d'une série de visualisations intégrées au site. Le passage au site web remet aussi en cause la hiérarchie habituelle entre ingénierie et recherche : la conception d'un site ne répond pas à une question scientifique, mais elle soulève ses propres questions. Loin d'être anodines, ces problématiques de design déterminent la construction et la réception des savoirs. Il est donc important, je pense, de problématiser ces questions de visualisation et de design.

Première partie

Du document numérisé au XML-TEI : nature du corpus, structure des documents et méthode de production des données

Chapitre 1

Le marché des manuscrits autographes au prisme des catalogues de vente

Ce chapitre présente l'objet d'étude du projet *MSS* : étudier le marché des manuscrits autographes du XIX$^{\text{ème}}$ s.. parisien à partir de ses catalogues de vente et étudier la construction du canon littéraire au prisme du marché du manuscrit.

1.1 Pourquoi étudier le marché des manuscrits autographes ?

Cette section porte sur l'intérêt scientifique des objets d'étude du projet (marché des manuscrits et étude de la construction du canon).

1.2 La structure du corpus : périodisation, producteurs des documents et classification

Ici est faite une présentation des documents traités dans le cadre du projet MSS. La présentation est à deux niveaux : au niveau du corpus et des catalogues. Ce chapitre s'appuie sur les mémoires effectués par d'ancien.ne.s stagiaires de *Katabase*, qui ont déjà beaucoup analysé la nature et les enjeux du corpus [1].

1. Lucie Rondeau du Noyer, *Encoder automatiquement des catalogues en XML-TEI. Principes, évaluation et application à la revue des autographes de la librairie Charavay*, Mémoire pour le diplôme de master "Technologies numériques appliquées à l'histoire", Paris, École nationale des Chartes, 2019, URL : `https://github.com/lairaines/M2TNAH` (visité le 13/06/2022) ; Caroline Corbières, *Du catalogue au fichier TEI. Création d'un workflow pour encoder automatiquement en XML-TEI des catalogues d'exposition.* Mémoire pour le diplôme de master "Technologies numériques appliquées à l'histoire", Paris, École nationale des Chartes, 2020, URL : `https://github.com/carolinecorbieres/Memoire_TNAH`

1.2.1 Le corpus de catalogues de vente de manuscrits

Ici est présenté le corpus : nature, quantité de documents (et d'entrées individuelles), dates, différentes classifications qui peuvent être faites (revues, catalogues de ventes aux enchères ou à prix fixes...).

1.2.2 Structure des catalogues

Ici sera présentée la structure des catalogues ; la structure de chaque page ne sera détaillée qu'à la partie suivante.

(visité le 13/06/2022) ; Juliette Janès, *Du catalogue papier au numérique. Une chaîne de traitement ouverte pour l'extraction d'information issue de documents structurés*, Mémoire pour le diplôme de master "Technologies numériques appliquées à l'histoire", Paris, École nationale des Chartes, 2021, URL : `https://github.com/Juliettejns/Memoire_TNAH` (visité le 13/06/2022).

Chapitre 2

Production des données : de l'OCR à la TEI

Cette partie s'attache autant à présenter le processus d'océrisation (qui est déjà bien établi et ne constitue pas le cœur de mon stage) que la structure des documents. Alors que le chapitre précédent s'intéresse aux catalogues dans leur ensemble, ici, on étudie le corpus au niveau de la page et de l'entrée individuelle. En effet, l'océrisation repose sur la segmentation, et donc sur l'établissement d'une structure « abstraite » d'une page (c'est-à-dire, d'un découpage de la page en zones).

2.1 Extraire le texte des imprimés

2.1.1 Comprendre la structure du document pour préparer l'édition numérique

Appréhender la structure de la page à l'aide de SegmOnto

La structure des catalogues est présentée au niveau de la page. L'ontologie SegmOnto [1] est utilisée, autant pour appréhender la structure de la page que pour exprimer cette structure de façon standardisée.

Description des entrées de catalogue : préparer l'édition TEI

Ici, la structure des catalogues est présentée au niveau de l'entrée, c'est à dire du lot mis en vente. C'est à autour de la structure des entrées qu'est construite l'édition XML-TEI. On s'intéresse à la structure des entrées individuelles à deux niveaux :

1. Kelly Christensen, Simon Gabay, Ariane Pinche et Jean-Baptiste Camps, « SegmOnto – A Controlled Vocabulary to Describe Historical Textual Sources », dans *Documents anciens et reconnaissance automatique des écritures manuscrites*, Paris : École nationale des Chartes, 2022.

— Au niveau intellectuel : quelles sont les différentes parties d'une entrée (titre, description du manuscrit, prix...).

— Au niveau « textuel » : quels sont les séparateurs, c'est à dire les éléments dans le texte qui permettent de séparer les pages de catalogue en entrées et les entrées en sous-éléments) correspondant à la structure intellectuelle décrite ci-dessus.

2.2 L'encodage des manuscrits en XML-TEI

2.2.1 Encoder les catalogues en TEI

Ici est présentée la représentation XML-TEI des catalogues de vente.

2.2.2 L'encodage en TEI : un processus sélectif qui réduit les significations du texte

Après une étape d'océrisation via *eScriptorium*, le texte extrait des PDF peut être exporté soit en texte brut, soit en XML Page ou Alto. Ces formats s'attachent à garder une relation entre le XML et le document numérisé (les zones de texte sont indiquées, chaque ligne est dans une balise...). Cependant, l'unité intellectuelle centrale à la suite du projet, ce n'est pas la page numérisée, mais l'entrée de catalogue. Un format plus complexe que le XML d'*eScriptorium* est donc nécessaire. Assez logiquement, la suite du projet s'appuie sur une traduction des catalogues en TEI. On s'intéresse autant à la structure des documents XML (quelles balises sont utilisées...) qu'à l'intérêt scientifique d'une édition numérique (balisage sémantique, possibilité de normaliser les informations grâce à des attributs).

L'édition numérique en XML-TEI des catalogues implique une certaine perte d'informations : l'intégralité des significations contenues dans les catalogues imprimés ne peut être traduite en TEI (la police, ou la qualité du papier, peuvent être documentés mais ne peuvent pas être reproduites). Ce genre de perte d'information a lieu, à différents degrés, dans la plupart des éditions TEI : ce format n'est pas un substitut des documents originels. Dans le projet *MSS / Katabase*, d'autres informations sont perdues : l'édition numérique n'est pas censée être une représentation exhaustive des catalogues. La TEI n'est pas utilisée comme un format de conservation, mais comme un format de traitement qui sera enrichi dans les différentes étapes. Afin de mesurer ce qui est conservé et ce qui est perdu du document originel, l'édition TEI sera analysée à la lumière de la « roue du texte » du philologue Patrick Sahle [2] qui modélise les significations plurielles d'un texte.

John Frow + Susan Pearce ?

2. Patrick Sahle, « Digital modelling. Modelling the digital edition », dans *Medieval and modern manuscript studies in the digital age*, London/Cambridge, 2016, URL : https://dixit.uni-koeln.de/wp-content/uploads/2015/04/Camp1-Patrick_Sahle_-_Digital_Modelling.pdf, p. 11.

Deuxième partie

Normalisation, enrichissements et extraction d'informations : une chaîne de traitement pour des données semi-structurées

Chapitre 3

Faire sens d'un corpus complexe : homogénéisation des données et extraction d'informations

3.1 Homogénéiser et normaliser un corpus complexe

Cette section s'intéresse à la manière dont les fichiers `TEI` sont traités afin de pouvoir ensuite en extraire des informations. C'est directement grâce la structure des entrées (et grâce à la nature « semi-structurée » des catalogues) qu'est possible le traitement automatisé des documents.

3.1.1 Pourquoi chercher à normaliser le corpus ?

La question mérite d'être posée : des étapes de post-traitement du corpus sont nécessaires pour pouvoir en extraire des informations et pour pouvoir donc en faire sens ; cependant ce processus peut également impacter la nature du texte et ses significations. Tout comme l'édition `TEI` originelle est un processus sélectif, le traitement des documents encodés est lui un processus sélectif : certaines informations contenues par l'encodage sont privilégiées aux dépens d'autres. Il s'agit ici d'expliciter ces choix (travailler sur les prix, les formats et dimensions des manuscrits plutôt que sur leur sujet) et de les justifier. Cette section s'attache donc à rappeler les questions de recherche qui sous-tendent la normalisation des documents (ajouter plus de structure au document `TEI` pour l'exploiter plus facilement, uniformiser la notation des tailles et des dimensions des documents...).

3.1.2 Comment normaliser le corpus tout en préservant sa valeur documentaire ?

Ici, on s'intéresse à la manière dont la TEI est mise à profit pour enrichir le corpus tout en conservant le contenu textuel des catalogues. Les différentes étapes de normalisation sont également rappelées (ce travail n'étant pas au cœur de mon stage, il s'agira plutôt d'un rappel que d'une présentation technique détaillée).

3.2 Faire sens du corpus : extraction d'informations et fouille de texte

Ici sont décrits le processus et les objectifs de l'extraction d'informations à partir des fichiers TEI. C'est à partir de cette opération d'extraction que sont construites les visualisations, qui permettent une approche graphique du corpus et une meilleure compréhension de celui-ci.

3.2.1 Extraire des informations au niveau des entrées

Des données sont extraites pour chaque entrée de catalogue (un travail largement effectué par A. Bartz, que j'ai légèrement mis à jour) : prix (dans la monnaie de l'époque et en francs constants), date de vente normalisée, nom de l'auteur.ice et description du manuscrit... L'extraction d'informations pour les entrées individuelles permet surtout de faire une réconciliation des manuscrits vendus (c'est à dire, de retrouver les items vendus plusieurs fois).

3.2.2 Extraire des informations au niveau des catalogues

Un second processus d'extraction produit des données pour chaque catalogue de vente : titre du catalogue, date de vente, nombre d'items vendus, prix minimum, inférieur et maximum, prix moyen et médian, variance... Ce processus met l'accent sur une approche statistique et économique du corpus, qui permettra d'étudier l'évolution du cours et du volume du marché des manuscrits (nombre d'items en vente, évolution des prix).

3.2.3 Vers une approche économique du corpus : la conversion automatique des prix en francs constants

En même temps que ces processus d'extraction d'informations, un script de conversion des monnaies (françaises et étrangères) en francs constants 1900 a été élaboré. En annulant l'effet de l'inflation, les francs constant permettent d'étudier l'évolution réelle des prix.

Chapitre 4

Vers une étude des facteurs déterminant le prix des documents : alignement des entrées du catalogue avec *Wikidata* et exploitation de données normalisées

Ce chapitre est construit autour d'une question de recherche : comment produire des informations exploitables pour une étude économétrique à partir d'un corpus textuel semi-structuré ? Un des objectifs du projet est de faire l'étude des facteurs déterminant le prix d'un manuscrit. Pour faire cette étude, il faut obtenir, pour chaque entrée du catalogue, un certain nombre d'informations normalisées. Le travail d'extraction de données présentes dans les catalogues a déjà été fait par de précédent.e.s stagiaires. Ces données sont principalement quantitatives : prix des manuscrits, dimensions et nombre de pages, date de création. Il est nécessaire de compléter les informations par des données qualitatives et d'enrichir les données disponibles avec des sources extérieures. Pour ce faire, il a été choisi d'aligner le nom des auteur.ice.s des manuscrits avec des identifiants *Wikidata* ; dès lors que l'on a un identifiant *Wikidata*, il est possible de récupérer automatiquement des informations sur les personnes via `SPARQL`. Le choix de travailler uniquement sur les noms, et non sur la description des documents, a deux motivations :

— Les noms de personnes (et la manière dont elles sont décrites) constituent la partie la plus normalisée des documents. La description des manuscrits est plutôt en « texte libre ». Dans la continuité avec le reste du projet, nous sommes resté dans une approche « basse technologie », qui consiste à s'appuyer majoritairement sur des solutions techniquement simples. C'est pourquoi nous avons préférer traiter les noms

avec des tables de correspondance [1] et des expressions régulières, plutôt que de faire du TAL sur la description des documents.

— Toutes les informations « simples » (données quantitatives facilement normalisables : dates etc.) ont déjà été extraites des descriptions des manuscrits.

Ce travail d'enrichissement a été fait en deux temps.

La première étape, et la plus difficile, est l'alignement avec *Wikidata*. Cela demande d'extraire un ensemble d'informations à partir du nom de la personne et de la description de celle-ci. Parmi les informations extraites : nom, prénom, titre de noblesse, occupation, dates de vie et de mort. À partir de ces informations, stockées dans un dictionnaire, un algorithme construit successivement différentes chaînes de caractères pour lancer des recherchers en plein texte sur l'`Interface de programmation d'application (API),` de *Wikidata*. L'objectif est que le premier résultat recherché sur *Wikidata* soit correct. Sur un jeu de test, le score F1 obtenu est de 68%. Un relecture « manuelle » des résultats est donc nécessaire.

La deuxième étape, nettement plus simple, consiste à lancer des requêtes *Wikidata* sur les identifiants récupérés afin de récupérer des informations sur les auteur.ice.s des manuscrits (cette partie du travail est encore en cours) pour enrichir nos données.

Une fois ce travail effectué, l'enrichissement des données à proprement parler est possible : les fichiers `TEI` sont mis à jour pour ajouter les identifiants *Wikidata*. Ainsi, il est possible de faire le lien entre les entrées de catalogues dans des fichiers `XML` et les données issues de requêtes `SPARQL`, stockées dans un `JSON`.

4.1 Questions introductives : pourquoi et comment s'aligner avec *Wikidata* ?

Cette section, introductive, répond à des questions évidentes mais essentielles : elles permettent de mettre au clair l'intérêt et les (multiples) difficultés dans l'alignement avec *Wikidata*.

4.1.1 Pourquoi s'aligner avec des identifiants *Wikidata* ?

L'alignement avec *Wikidata* a pour objectif principal de mieux comprendre les déterminants du prix des manuscrits sur le marché du XIX$^{\text{ème}}$ s.. Mais pourquoi passer par un alignement avec *Wikidata* ?

Pour étudier les déterminants du prix d'un manuscrit, il faut établir la relation entre la variable dont la valeur est étudiée (le prix d'un manuscrit) et un ensemble d'autres fac-

1. C'est à dire, des tables qui permettent de normaliser la manière dont les informations figurent dans les catalogues, et donc de remplacer des termes « vernaculaires » par leurs équivalents utilisés par *Wikidata*

teurs (qui a écrit un manuscrit, quelles sont ses dimensions, de quand date le document...). En d'autres termes, il faut étudier le comportement d'une variable en fonction d'autres variables. En économétrie, cette opération s'appelle le calcul de régressions linéaires. La variable étudiée (le prix) est dite la variable expliquée ; les facteurs déterminant la valeur de cette variable sont dites « variables explicatives » [2]. Cependant, cette opération est loin d'être anodine : il faut d'abord identifier les variables pertinentes, et ensuite trouver un moyen de les quantifier. Deux difficultés se présentent pour alors.

Premièrement, il faut pouvoir quantifier les variables expliquées pour calculer des régressions linéaires. Il est possible de leur assigner une valeur numéraire (ce qui est aisé pour les informations quantitatives des catalogues : la date de l'écriture d'un manuscrit, ses les dimensions). Une autre possibilité est de quantifier la présence ou non d'une variable : mention d'un.e destinataire ou du contenu d'un manuscrit. Cependant, ces approches quantitatives ne permettent pas de quantifier des informations complexes, comme la célébrité des auteur.ice.s, ou encore si un manuscrit porte sur un évènement historique ou biographique important (le manuscrit d'un texte célèbre, par exemple, pourrait avoir une valeur particulières). Ces informations sont parfois être présentes dans les catalogues ; elles peuvent aussi être connues des lecteur.ice.s d'aujourd'hui et des acheteurs et acheteuses de l'époque. Il n'existe cependant pas de méthodes faciles pour détecter ou quantifier la célébrité d'une personne, ou l'importance d'un sujet.

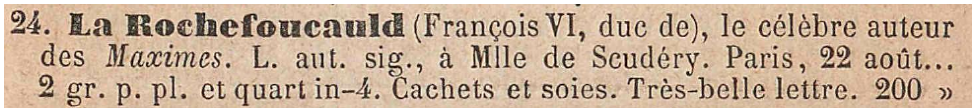
Une deuxième difficulté découle justement de la part d'implicite qu'il y a dans les catalogues. Les descriptions des items vendus sont brèves, et comprendre ce qui fait la valeur d'un manuscrit demande aux acheteur.euse.s d'avoir des références culturelles et historiques : celles-ci permettent d'identifier l'auteur.ice ou le sujet, et donc pour comprendre la valeur d'un manuscrit. Dans le cadre du projet *MSS / Katabase*, les entrées de catalogues sont traitées par une machine qui, en toute logique, ne dispose pas de ces références. La compréhension qualitative des entrées de catalogues n'est donc pas compatible avec l'approche par lecture distante du projet. Pour éviter de perdre ces informations qualitatives essentielles, il est donc nécessaire de trouver un moyen de quantifier le qualitatif.

En bref, la question est : comment faire la différence entre une lettre de La Rochefoucauld (4.1a), vendue 200 francs, et la deuxième (4.1b), vendue à 30 francs ? Le problème est un problème de lecture. Une observation de la description des lettres par un être humain comme par une machine peuvent identifier des éléments semblables dans le texte : les deux lettres sont écrites par des ducs ; l'une est une « Très-belle lettre » (4.1a), l'autre est une « Lettre intéressante » (4.1b) [3]. Bien que les lettres partagent des attributs, il y a une forte différence de prix entre les deux manuscrits. Un.e lecteur.ice peut trouver une raison à cette différence de prix : La Rochefoucauld et Madeleine de Scudéry n'ont pas

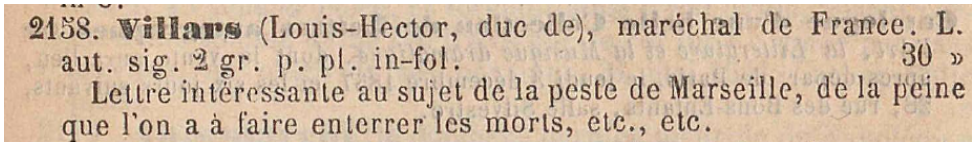
2. *Régression linéaire*, Wikipedia. L'encyclopédie libre, 2022, URL : `https://fr.wikipedia.org/wiki/R%C3%A9gression_lin%C3%A9aire` (visité le 10/07/2022).

3. Ce sont des informations qui se retrouvent souvent, et il est donc possible d'écrire un programme qui les relève automatiquement

le même statut que le duc de Villars. Un regard humain peut donc interpréter un prix et déterminer une valeur en s'appuyant sur ses connaissances. La lecture est qualitative et s'appuie sur de l'implicite, ce qui n'est pas possible pour une machine : formellement, rien ne distingue un nom d'un autre ; lorsqu'un programme « lit » un texte, il ne peut pas s'appuyer sur ses connaissances pour déterminer ce qu'un nom signifie, ce à quoi il fait référence.


24. **La Rochefoucauld** (François VI, duc de), le célèbre auteur des *Maximes*. L. aut. sig., à Mlle de Scudéry. Paris, 22 août... 2 gr. p. pl. et quart in-4. Cachets et soies. Très-belle lettre. 200 »

(a) Une lettre écrite par La Rochefoucauld vendue à 200 francs.


2158. **Villars** (Louis-Hector, duc de), maréchal de France. L. aut. sig. 2 gr. p. pl. in-fol. 30 »
Lettre intéressante au sujet de la peste de Marseille, de la peine que l'on a à faire enterrer les morts, etc., etc.

(b) Une lettre écrite par Louis-Hector Villars vendue à 30 francs.

FIGURE 4.1 – Deux exemples de lettres

Pour analyser efficacement la variable « prix », il faut pourtant pouvoir, dans une certaine mesure, comprendre les informations implicites et qualitatives contenues dans les catalogues. Le parti pris a donc été de construire le socle de connaissance qui manque à une machine, en s'alignant avec *Wikidata* et en s'en servant pour enrichir nos données. Le choix a été fait de ne s'aligner avec *Wikidata* que pour certaines parties des entrées de catalogue. Pour rappel, voici leur structure (4.1) :

Les entrées de catalogue contiennent beaucoup d'informations qualitatives, qui pourraient avoir une influence sur le prix du manuscrit : ici par exemple, la description du contenu de la lettre dans le `note` ; il est également souvent fait mention du ou de la destinataire. Cependant, l'alignement avec *Wikidata* n'a pas été fait avec l'intégralité des entrées. C'est seulement le contenu du `name` qui a été aligné avec *Wikidata*, à l'aide des informations contenues dans le `trait`. Le `desc` a déjà fait l'objet d'un grand travail de normalisation et d'extraction d'informations ; un alignement avec des sources externes n'aurait donc pas une très grande plus-value. L'élément `note` contient souvent des informations intéressantes, puisque c'est là qu'est décrit le contenu d'un manuscrit. Cependant, cet élément n'est pas toujours présent ; son contenu est souvent écrit en langage naturel, non structuré, et contient des informations trop variées pour développer un traitement uniforme. Il est donc difficile de tirer parti de cet élément. Le `name` et son `trait` sont les éléments les plus régulièrement présents ; les informations qu'ils contiennent sont toujours les mêmes (nom d'une personne ou thème d'un manuscrit dans le `name`, description du `name` dans le `trait`) ; enfin, ces deux éléments n'ont pas du tout été transformés dans le reste de la chaîne de traitement. Ils portent donc des informations qualitatives centrales pour produire des données exploitables dans une étude économétrique.

```xml
<item n="287" xml:id="CAT_000126_e287">
    <num type="lot">287</num>
    <name type="author">Tascher de la Pagerie
    ↪ (Marie-Euphémie-Désirée)</name>
    <trait>
        <p>fille de Joseph Tascher qui passa à Saint-Domingue, et de
        ↪ Mlle de la Chevalerie; elle épousa M. de Renaudin, puis le
        ↪ marquis François de Beauharnais et contribua beaucoup à la
        ↪ fortune de la future impératrice Joséphine.</p>
    </trait>
    <desc xml:id="CAT_000126_e287_d1">
        Pièce originale;
        (<date when="1780">1780</date>),
        <measure type="length" unit="p" n="4">4 p.</measure>
        <measure type="format" unit="f"
        ↪ ana="#document_format_4">in-4.</measure>
    </desc>
    <measure commodity="currency" unit="FRF" quantity="15">15</measure>
    <note>Plaidoirie pour le dit Renaudin contre sa femme qu'on accusait
    ↪ d'avoir usé de l'influence de son parent, M. de Beauharnais,
    ↪ pour se faire épouser</note>
</item>
```

Code source 4.1 – Représentation XML-TEI d'une entrée de catalogue

Le parti pris a donc été d'aligner avec des identifiants *Wikidata* les noms contenus dans les balises `name` à l'aide des descriptions contenues dans les `trait` ; à partir de cet alignement a été constituée une base de données. Cela permet d'approximer une lecture « humaine » des items en vente : pour chaque auteur.ice, un certain nombre d'informations auront été récupérées pour mieux identifier la personne (ses occupations, son origine, ses dates de vie...). L'analyse du corpus s'appuit alors sur un bagage de connaissances qui permet d'appréhender par lecture distante l'importance d'une personne. Il devient alors envisageable de voir dans quelle mesure la mention d'une personne impacte le prix d'un manuscrit, et quels sont les facteurs biographiques déterminant dans l'établissement de la valeur. Pour revenir à l'exemple de La Rochefoucauld : à défaut de permettre de savoir qui il est, un alignement avec *Wikidata* permet d'identifier son statut et sa place dans la culture française, en récupérant le nombre de ses publications ou encore les institutions dont il est membre.

4.1.2 Présentation générale de l'algorithme

Construire un jeu de données issu de `SPARQL` à partir de la manière dont une personne est nommée et décrite au XIX$^{\text{ème}}$ s. n'est pas une opération anodine. La chaîne de traitement est donc assez complexe, comme le montre le schéma 4.2. Cette chaîne de traitement peut être séparée en trois étapes.

Étape 1 – Extraction et structuration de données

Premièrement, il s'agit d'aligner les entrées de catalogue avec des identifiants *Wikidata*. Ceux-ci sont liés à des « entités » *Wikidata* : des personnes, lieux et évènements décrits dans *Wikidata* par un certain nombre de propriétés (date de naissance, lieu de résidences...). Cette première étape repose avant tout sur l'extraction et la traduction des données depuis les éléments `name` et `trait`. Ce processus d'extraction permet de récupérer toutes les données pertinentes pour chaque entrée de catalogue et de les stocker dans un dictionnaire structuré. Comme on le verra, la nature « semi-structurée » des entrées (ainsi qu'une bonne connaissance du corpus) permet de d'automatiser le processus d'extraction et de traduction des données par détection de motifs, sans avoir à passer par l'apprentissage machine : étant donné que les mêmes types d'informations sont toujours présentes et que les entrées suivent des modèles relativement proches, il est possible de s'appuyer sur la structure des entrées pour identifier les informations pertinentes. L'extraction de données repose donc sur de la détection de motifs à l'aide d'expressions régulières : des récurrences sont repérées dans le texte et utilisées pour distinguer différentes informations (nom, prénom, titre de noblesse...). Pour appuyer l'usage d'expressions régulières par une méthode plus « qualitative » et précise, certains termes particuliers sont extraits et éventuellement traduits à l'aide de tables de conversion (c'est-à-dire de dictionnaires

qui associent à un terme dans le texte une version normalisée).

Étape 2 – récupération d'identifiants *Wikidata* via des recherches en plein texte à l'aide d'une `API`

Une fois les données du `name` et du `trait` structurées en dictionnaire, elles sont utilisées pour lancer plusieurs recherches en plein texte sur le moteur de recherche de *Wikidata*. Ces recherches sont faites automatiquement grâce à l'`API` de *Wikidata*. Pour maximiser les chances d'obtenir un identifiant valide, un algorithme a été conçu pour lancer plusieurs recherches à partir de chaque dictionnaire. La première recherche met bout-à-bout toutes les valeurs disponibles dans le dictionnaire. Ensuite, en fonction des paramètres de recherche disponibles dans le dictionnaire, différentes autres recherches sont lancées. Cet algorithme a été élaboré en menant de nombreux tests pour maximiser le taux de réussite, calculé sous la forme d'un score F1.

Étape 3 – constitution d'un jeu de données à l'aide de `SPARQL`

Si l'étape d'alignement avec *Wikidata* est la plus complexe, elle n'est qu'une étape préparatoire vers la constitution du jeu de données. En fait, récupérer les identifiants est seulement ce qui rend possible l'enrichissement en tant que tel : en lançant une requête `SPARQL` sur tous ces identifiants, il est possible, pour chaque entité représentée par l'identifiant, de récupérer des informations depuis *Wikidata* et donc de construire le jeu de données définitif. Pour cette étape, le processus est plus simple : les identifiants récupérés à la fin de l'étape précédente sont dédoublonnés (pour éviter de lancer plusieurs fois la même requête) ; ensuite une requête `SPARQL` est initalisée et lancée chacun des identifiants. Les résultats sont traduits depuis les formats `JSON` ou `XML` retournés par `SPARQL` sous forme de `JSON` plus simple, et donc plus aisément manipulable. Le jeu de données est enregistré dans un fichier. Pour finir, les identifiants *Wikidata* sont réinjectés aux catalogues `TEI`, afin de pouvoir faire le lien entre les catalogues et le jeu de données qui a été construit.

Cette chaîne de traitement étant lancée sur plus de 80000 entrées de catalogues, le temps d'exécution est très long et même des petites améliorations de performance peuvent avoir un grand impact ; dans sa version initiale, le script demandait des performances particulièrement élevées, et ne fonctionnait pas sur un ordinateur aux capacités limitées. La chaîne de traitement a donc été reprise en plusieurs points afin d'être optimisée, de fonctionner plus vite en étant moins coûteuse en ressources.

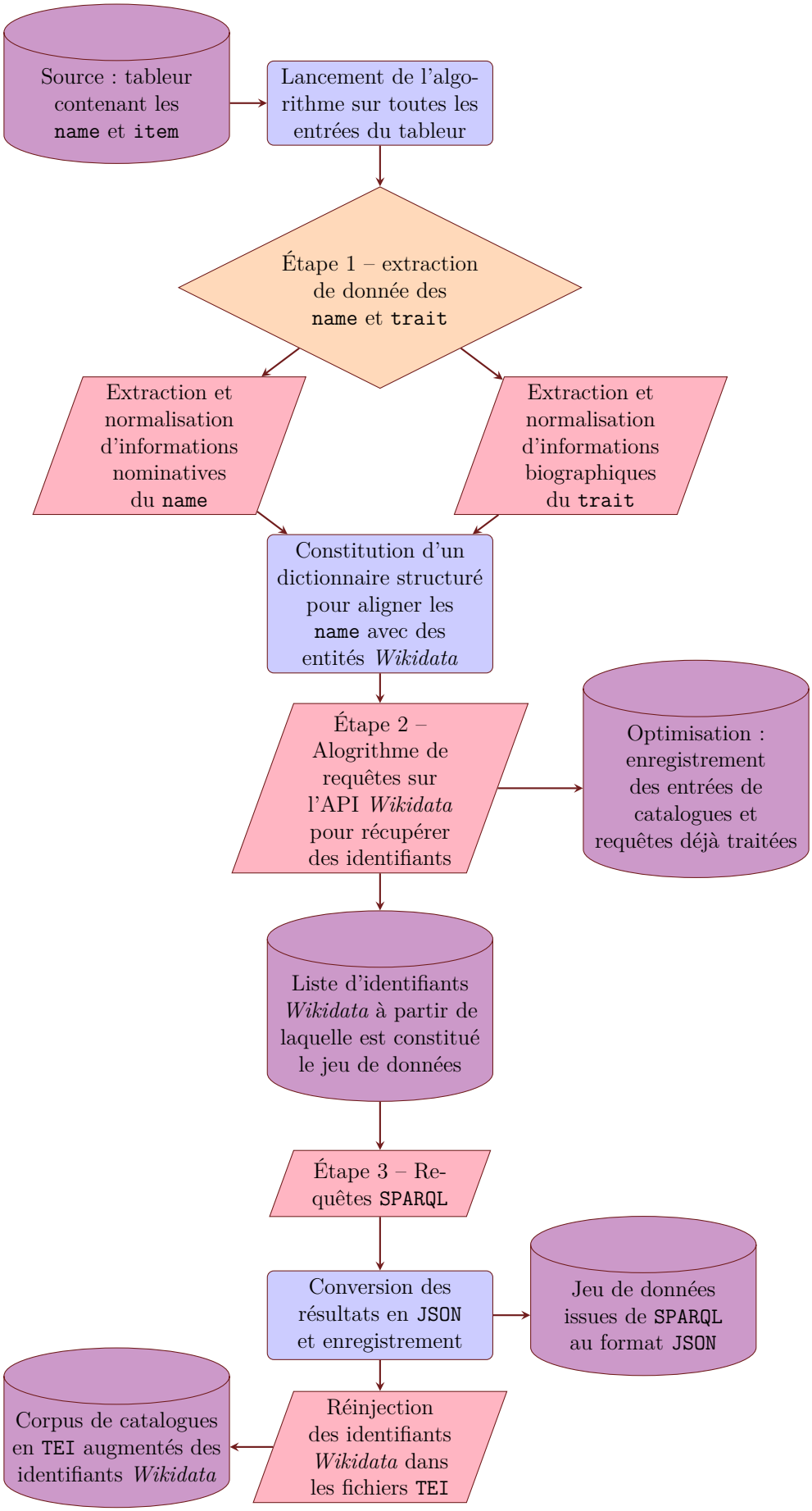


FIGURE 4.2 –
Présentation générale de l'algorithme d'enrichissement de données à l'aide de *Wikidata*

4.1.3 Comment traduire des descriptions textuelles datant du XIX^{ème} s. en chaînes de caractères qui puissent retourner un résultat sur *Wikidata* ?

Dans la réalisation de cet algorithme, la principale difficulté porte sur la récupération d'identifiants *Wikidata* à l'aide de recherches en plein texte. Le script prend en entrée un nom et sa description – tels qu'elles figurent dans des catalogues datant majoritairement du XIX^{ème} s.. La difficulté, au delà de la détection et de l'extraction d'informations, est de traduire ces informations pour qu'elles permettent de trouver des résultats pertinents sur *Wikidata*. Ce problème est autant linguistique de technique. Une personne ou une chose est nommée ou décrite d'une certaine manière dans un catalogue de vente ancien. Il n'y a aucune garantie que cette caractérisation corresponde à celle faite par *Wikidata* : l'orthographe des noms évoluent, tout comme la manière de nommer certains métiers. À ces évolutions orthographiques s'ajoutent des évolutions intellectuelles : les titres de noblesse sont un marqueur plus important au XIX^{ème} s. français que dans un XXI^{ème} s. mondialisé. Une personne n'est que rarement décrite par son titre dans *Wikidata*.

Le problème de la traduction des noms

Il existe bien sûr des cas simples, comme l'exemple 4.2 : en extrayant le contenu du `name`et en traduisant le « roi » issu du `trait`, la chaîne de caractère obtenue est « Henri IV king ». En recherchant cette chaîne de caractère sur *Wikidata*, le premier résultat obtenu est correct. Cependant, de nombreux cas sont plus complexes, surtout lorsque l'auteur.ice du manuscrit est moins célèbre. L'exemple 4.3 est éclairant : dans le catalogue, la personne est nomée « Bruno Daru » ; sur *Wikidata*, le nom de la personne est « Pierre Daru », et son nom complet Pierre Antoine Noël Bruno Daru. Si la recherche en plein texte est faite avec les mêmes paramètres que pour l'exemple précédent (nom de la personne et titre de noblesse), le premier résultat obtenu n'est pas le bon : c'est un renvoi à un article de *l'Encyclopedia Britannica* datant de 1911. C'est en cherchant seulement le nom est le prénom que *Wikidata* retourne un résultat pertinent. Il est intéressant de retenir deux choses de cet exemple : dans les catalogues, le prénom d'une personne correspond en fait souvent à son deuxième ou troisième prénom ; ensuite, le titre de noblesse est un critère plus fréquemment mentionné dans les catalogues que dans *Wikidata*. Cela s'explique assez aisément : le XIX^{ème} s. connaît une alternance de régimes politiques (royauté, empire, république) où la noblesse n'a pas encore perdu son pouvoir. La probabilité qu'un titre de noblesse soit mentionné sur *Wikidata* diminue lorsqu'un titre est peu important ; dans les catalogues, cependant, même les titres les moins importants sont régulièrement mentionnés. Par conséquent, seuls les titres les plus importants seront extraits pour lancer une recherche sur l'`API` de *Wikidata*.

Dans le cas de noms de personnes étrangères, la situation peut être plus complexe

```xml
<item n="134" xml:id="CAT_000233_e134">
    <!-- ... -->
    <name type="author">Henri IV</name>
    <trait>
        <p>roi de France.</p>
    </trait>
    <!-- ... -->
</item>
```

Code source 4.2 – Un cas simple : Henri IV roi de France

```xml
<item n="98" xml:id="CAT_000082_e98">
    <!-- ... -->
    <name type="author">Daru (Bruno, comte)</name>
    <trait>
        <p>célèbre ministre de Napoléon Ier, historien de Venise, de
        ↪ l'Acad. fr., né à Montpellier</p>
    </trait>
    <!-- ... -->
</item>
```

Code source 4.3 – Un cas plus complexe : Pierre Antoine Noël Bruno Daru

encore. L'exemple 4.4 combine différentes difficultés.

— D'abord, la personne est étrangère ; dans les catalogues, les noms sont systématiquement francisés – « Albert-Venceslas-Eusèbe » dans le catalogue, « Albrecht Wenzel Eusebius » en langue originelle. Se pose donc la question de si le nom doit être traduit, et si oui comment ?

— Ensuite, comme l'indique la présence de « dit » dans le `name`, il est mentionné un nom de naissance (« de Waldstein ») et un nom d'usage (« Wallenstein »). Idéalement, il faudrait choisir entre l'un ou l'autre, plutôt que de rechercher « Waldstein Wallenstein » sur *Wikidata*, ce qui risque d'augmenter le bruit.

Notre approche s'appuyant sur la structure du texte, le deuxième point peut être réglé : le nom d'usage est écrit au début, et le nom de naissance entre parenthèses (c'est également le cas des noms de personnes nobles, par exemple). Il est donc possible de choisir l'un ou l'autre nom. Le premier point est plus problématique : si la traduction du nom serait envisageable en théorie, celle-ci est difficilement compatible avec une approche basée sur la détection de motifs dans le texte : le prénom est repérable comme étant un motif (trois noms séparés par des tirets) ; cependant, il est impossible de le traduire automatiquement (ce qui demanderait de connaître la langue dans laquelle un prénom doit être traduit). C'est ici que les informations contenues dans le `trait` prennent leur

```xml
<item n="5518" xml:id="CAT_000401_e5518">
    <!-- ... -->
    <name type="author">Wallenstein (Albert-Venceslas-Eusèbe de
    ↪ Waldstein dit)</name>
    <trait>
        <p>duc de Friedland, célèbre général de la guerre de Trente ans.
        ↪ Assassiné en 1634.</p>
    </trait>
    <!-- ... -->
</item>
```

Code source 4.4 – Le problème des noms de personnes étrangères

importance : lorsqu'il y a des défaillances dans les informations nominatives, des données biographiques permettent de diminuer le risque d'erreurs. Dans cet exemple, recherche « Albert-Venceslas-Eusèbe Waldstein » ne retourne aucun résultat, de même que rechercher Albert-Venceslas-Eusèbe Wallenstein. Cependant, le bon résultat est obtenu en recherchant « Wallenstein 1634 ». Une difficulté supplémentaire vient avec ce type de cas : différents paramètres de recherche (nom, prénom...) ont un impact différent dans l'obtention du bon résultat en fonction des personnes sur qui la requête est faite. Dans ce cas, rechercher le nom d'usage et la date de naissance retourne un résultat valide, ce qui n'est pas toujours le cas. Pour contourner ce problème, trois solutions ont été mises en place : d'abord, ce types de requêtes a été fait « à la main », de façon non-automatique, pour de nombreuses entrées différentes afin de déterminer la meilleure combinaison de caractères ; ensuite, des tests qui permettent de mesurer l'influence de chaque paramètre de recherche dans l'obtention du résultat ; enfin, l'algorithme final lance successivement différentes requêtes avec différents paramètres afin de maximiser la probabilité d'obtenir un résultat valide. Nous reviendrons plus en détail sur les deux derniers points.

L'extraction d'informations biographiques : une autre difficulté

Cependant, le problème ne s'arrête pas qu'aux noms. Dans un exemple ; précédent, le titre de noblesse influençait l'obtention d'un résultat valide. De nombreuses autres informations biographiques pourraient, au premier abord, permettre d'obtenir le bon résultat. C'est souvent le cas, puisque extraire le métier ou la fonction d'une personne permet de supprimer les faux positifs retournés par l'`API`. C'est par exemple le cas dans l'exemple 4.5. En cherchant uniquement le nom et le prénom (« Hans Bulow »), le premier résultat retourné renvoie à un journaliste suédois. Extraire le mot « pianiste » `trait` et le traduit en anglais permet d'obtenir le bon résultat.

L'extraction d'informations biographiques et leur utilisation dans des requêtes est donc pertinent. Cependant, extraire trop d'informations conduit à lancer des requêtes qui

```xml
<item n="136" xml:id="CAT_000189_e136">
    <!-- .. -->
    <name type="author">Bulow (Hans)</name>
    <trait>
        <p>le célèbre pianiste.</p>
    </trait>
</item>
```

Code source 4.5 – Un exemple où l'extraction du métier permet l'obtention du bon résultat

ne renvoient aucun résultat. Dans les exemples 4.6 et 4.7, extraire et traduire des fonctions conduit à lancer les requêtes « John Okey colonel » et « Jean Bouhier president » qui ne retournent aucun résultat, ou des résultats qui ne sont pas valides. Cependant, dans les deux cas, si une requête est lancée sans la fonction, un résultat correct est obtenu. Les raisons pour lesquelles des résultats erronés sont retournés ne sont cependant pas les mêmes, et il est intéressant de mieux observer les requêtes lancées et les résultats obtenus. Dans le premier cas, le terme mis en avant dans le `trait` (« colonel ») n'est pas celui avec lequel la personne est décrite sur *Wikidata* (où John Okey est décrit comme étant un homme politique). Cela met en avant un problème relatif au changement de regard sur des personnalités : dans un contexte, la personne est décrite comme une figure militaire, dans l'autre comme une figure politique. Le deuxième cas est plus technique. Il y a en fait une erreur dans la requête qui ne retourne pas de résultat (« Jean Bouhier president ») : un.e président.e de parlement n'est en général pas décrite comme « président ». Cependant, en extrayant des données uniquement par détection de motifs, il est possible de repérer et traduire un terme générique comme « président ». Extraire le complément « Parlement de Dijon » du `trait` n'est cependant pas possible (cela impliquerait d'étudier la grammaire de la phrase, pour mettre en avant la relation entre « président » et « Parlement de Dijon »). Au vu de la taille et de la variété du jeu de données, il est impossible de traiter au cas par cas des entrées, ou préciser la détection de motif avec suffisamment de précision pour pouvoir résoudre ce genre de difficultés.

De situations comme les exemples 4.6 et 4.7, il faut donc retenir que l'extraction d'informations vient nécessairement avec un risque d'erreur. Le parti pris a donc été de ne pas repérer les métiers et autres termes très spécifiques, comme les grades militaires et les titres de noblesse peu élevés : ils ne retournent pas de résultats sur le moteur de recherche. Ensuite, plus des requêtes sont précises, plus elles risquent de retourner du silence (c'est-à-dire, de ne pas donner de réponse) ; cependant, si un résultat est obtenu, il est plus probable que ce résultat soit correct. Une fois l'extraction d'informations faite, l'algorithme d'extraction d'identifiants sur l'`API` *Wikidata* a donc été conçu en suivant un principe soustractif : les premières recherches sont faites avec un maximum de paramètres ; si aucun résultat n'est obtenu, des paramètres sont enlevés pour que l'`API` retourne un

```xml
<item n="152" xml:id="CAT_000189_e152">
    <!-- ... -->
    <name type="author">Okey (John)</name>
    <trait>
        <p>colonel anglais, un des lieutenants de Cromwell.</p>
    </trait>
    <!-- ... -->
</item>
```

Code source 4.6 – Quand l'extraction d'un métier conduit à des requêtes trop spécifiques

```xml
<item n="5430" xml:id="CAT_000401_e5430">
    <!-- ... -->
    <name type="author">Bouhier (Jean)</name>
    <trait>
        <p>président au Parlement de Dijon, membre de l'Académie
        ↪ française.</p>
    </trait>
    <!-- ... -->
</item>
```

Code source 4.7 – Le cas des métiers dont l'extraction est problématique

plus grand nombre de résultats. Enfin, ces deux exemples montrent qu'il n'est pas possible d'extraire et de traduire des informations sans prendre en compte ce qui sera pertinent pour le moteur de recherche de *Wikidata*. Il ne s'agit donc pas seulement d'extraire des informations, mais aussi de s'adapter avec ce moteur de recherche pour augmenter la probabilité d'obtenir un résultat valide.

4.1.4 Comment négocier avec le moteur de recherche de *Wikidata* ?

Comme cela commence à apparaître, l'extraction d'informations, lorsqu'elle vise à interagir avec des données externes, vient avec des difficultés supplémentaires. Il ne faut pas seulement extraire les informations ; leur extraction et structuration doivent permettre de lancer des recherches en plein texte, et donc de minimiser le bruit (les informations non pertinentes) et le silence (l'absence d'informations) de la part du moteur de recherche. Il faut donc traduire les informations extraites pour qu'elles correspondent au vocabulaire utilisé par *Wikidata*. Cette opération n'est pas anodine : si les catalogues de vente fonctionnent avec leurs propres catégories, le même peut être dit de *Wikidata* : certains types de données sont plus souvent référencées que d'autres et *Wikidata* utilise un vocabulaire qui lui est propre. Pour bien mener ce processus de traduction et de

structuration de l'information, il est nécessaire de bien connaître le fonctionnement de ce moteur de recherche pour mieux s'y adapter.

Comme cela a été dit, l'alignement avec *Wikidata* passe par l'utilisation de l'`API` mise en point par l'institution afin de lancer automatiquement des recherches en plein texte ; l'objectif est que le premier résultat retourné par le moteur de recherche soit le bon. La première chose à remarquer est que, contrairement à un moteur de recherche généraliste (comme *Google, QWant...*), ce moteur n'est pas compatible avec des requêtes approximatives. L'exemple 4.8 est pertinent à ce égard [4]. Dans de nombreuses entrées, comme c'est le cas ici, les fonctions d'une personne ayant participé à la révolution sont présentées de façon précise : Marc David Alba Lasource est décrit comme étant un « conventionnel girondin ». Cette mention, régulièrement présente dans les catalogues, pourrait être relevée en tant que telle. Cependant, lancer la recherche « Lasource conventionnel » ne retourne aucun résultat. Si la même recherche est lancée sur un moteur de recherche généraliste (ici, *QWant*), la page *Wikipedia* de Lasource fait partie des premiers résultats [5]. Cette différence dans les données retournées par les moteurs de recherche a deux explications : un moteur de recherche généraliste recherche les occurrences de mots, non seulement dans le titre de la page, mais aussi dans le corps du texte. Si le mot « conventionnel » est absent du titre, il est certainement à plusieurs reprises dans une notice biographique type *Wikipedia*. *Wikidata* ne contenant que des données, et pas de texte en tant que tel, l'indexation du corps du texte par le moteur de recherche interne à *Wikidata* n'est pas possible. Ensuite, la plupart des moteurs de recherche généralistes utilisent des méthodes de traitement du langage afin de simplifier la requête lancée par l'utilisateur.ice : les mots recherchés sont simplifiés, le moteur de recherche associe les termes recherchés avec d'autres termes « cooccurents », c'est-à-dire fréquemment utilisés ensemble [6]. Dans le cas du moteur de recherche de *Wikidata*, la requête de l'utilisateur.ice ne semble pas être retraitée : des signes de ponctuation ou des fautes de frappes influencent l'obtention d'un résultat, de même que l'usage de termes inadaptés.

Pour faire face à la « rigidité » relative du moteur de recherche de *Wikidata*, il est donc nécéssaire de préparer ses données au moment de leur extraction. En prenant le même

4. Dans cet exemple, le prénom, « M.-D.-A. », n'est pas pris en compte pour se concentrer sur l'utilisation d'informations biographiques dans le `trait`.

5. Le 29/07/2022, c'est le troisième résultat ; le premier correspond à une vente aux enchères d'archives du conventionnel. Les moteurs de recherche pouvant être mis à jour régulièrement, il est possible que l'ordre des résultats change

6. *Moteur de recherche*, Wikipedia. L'encyclopédie libre, 2022, URL : `https://fr.wikipedia.org/wiki/Moteur_de_recherche` (visité le 17/07/2022). Pour des analyses plus détaillées sur la construction d'ensemble de termes coocurrents via le développement de vecteurs de mots, voir Tomas Mikolov, Kai Chen, Greg Corrado et Jeffrey Dean, « Efficient Estimation of Word Representations in Vector Space » (, 2013), Publisher : arXiv Version Number : 3, DOI : `10.48550/ARXIV.1301.3781` ; pour un article technique détaillant la classification et la sélection de résultats pertinents par apprentissage profond, voir Paul Covington, Jay Adams et Erme Sargin, « Deep Neural Networks for YouTube Recommendations », dans *Proceedings of the 10th ACM Conference on Recommender Systems*, New York, 2016, URL : `https://research.google/pubs/pub45530/` (visité le 10/06/2022)

```xml
<item n="140" xml:id="CAT_000197_e140">
    <!-- ... -->
    <name type="author">Lasource (M.-D.-A.)</name>
    <trait>
        <p>célèbre conventionnel girondin, né près de Montpellier en
        ↪ 1762, guillotiné en 1793.</p>
    </trait>
    <!-- ... -->
</item>
```

Code source 4.8 – Le problème de l'approximation et de la traduction : Lasource, conventionnel

exemple (4.8), un résultat correct peut être obtenu en remplaçant « conventionnel » par « politician » [7], pour rechercher sur *Wikidata* « Lasource politician ». Ici, la traduction de « conventionnel » en « politician » est d'autant plus intéressante que la date de naissance dans le catalogue (1762) ne correspond pas à celle indiquée sur *Wikidata* (1763). Dans un cas comme celui-ci, où certaines données sont incorrectes, il est important d'extraire un maximum d'informations pour que, si certaines requêtes ne rapportent pas de résultats, pouvoir en faire d'autres avec différents paramètres.

En conclusion, il faut retenir que le moteur de recherche de *Wikidata* n'admet pas d'erreurs, ni de requêtes partiellement erronées (dans l'exemple 4.8, où la date de naissance soit correcte, mais pas la date de mort) ; il ne prend pas non plus en compte la synonymie, ce qui veut dire qu'il n'améliore pas la requête lancée par un.e utilisateur.ice... Cela signifie que les termes utilisés dans une requête doivent être adaptés à ceux que *Wikidata* utilise. Les termes spécifiques utilisés dans les catalogues (« conventionnel »), mais aussi de nombreux titres militaires et de noblesse peu élevés (« capitaine », « marquis ») sont relativement rarement présents sur *Wikidata*. Lorsque les requêtes sont lancées, de tels termes sont donc abandonnés et parfois remplacés par des termes plus génériques : par exemple, « capitaine » est remplacé par « military », traduction anglaise de « militaire ». De même, des termes principalement en usage dans la langue française, comme « conventionnel » sont moins efficaces pour lancer des recherches.

4.1.5 Une approche prédictive

L'alignement avec *Wikidata* et l'extraction d'entités n'est donc pas une opération anodine : les données contenues dans les catalogues sont variées, autant par leur structure que par les informations qu'elles contiennent ; il peut être difficile à faire la traduction de données du XIX^{ème} s. en chaînes de caractères pouvant retourner des réponses valides sur *Wikidata* ; enfin, le l'alignement repose sur une bonne connaissance du moteur de

7. « Personnalité politique »

```
1 <name type="author">Verneuil (Charlotte Séguier duchesse de)</name>
```

Code source 4.9 – Peut-on identifier les différents éléments d'une phrase par détection de motifs ?

recherche de *Wikidata*.

De plus, la technique utilisée dans l'extraction de données, reposant sur la détection de motifs à l'aide de expressions régulières et de tables de conversion, est une technique qui vient avec un certain nombre d'incertitudes. Avec ce genre de techniques, il est impossible de « comprendre » ce qu'un élément signifie. Dans l'exemple 4.9, formellement, rien ne sépare le nom propre de la duchesse (« Séguier ») du nom de son duché (« Verneuil »). En s'appuyant sur une connaissance de la structure répétitive des entrées, il est uniquement possible de supposer que le nom entre parenthèses est un nom propre, tandis que le nom hors des parenthèses correspond au nom du duché. En bref, les méthodes de détection de motifs utilisées, peuvent uniquement inférer le sens d'un mot par rapport à sa position dans une phrase. Si cette technique implique une certaine incertitude, elle est cependant particulièrement adaptée à un corpus semi-structuré, comme c'est le cas des catalogues de vente de manuscrits, et à l'opération d'alignement avec *Wikidata*. Comme cela est expliqué plus bas, au fond, il n'est pas tellement important de distinguer le sens des différentes informations : ce qui a du sens, c'est que l'extraction et la structuration des informations permet de construire des chaînes de caractères à rechercher sur *Wikidata*. Identifier la fonction d'un mot n'est donc ici qu'un moyen – contrairement à de l'analyse lexicale, où la fonction des mots dans une phrase est signifiante –. En effet, repérer le rôle que tiennent les termes extraits (métier, prénom...) permet de mieux construire la chaîne de caractère recherchée sur *Wikidata*, en pouvant filtrer certaines informations (retirer les dates de vie et de mort, par exemple).

Étant donnée cette quantité d'incertitudes, l'approche suivie dans l'alignement avec *Wikidata* peut être qualifiée de « prédictive ». Par ce terme, il faut comprendre que il n'y a pas, de certitude totale dans le processus d'extraction et de traduction des données. Il n'est pas possible de récupérer avec une certitude totale le bon identifiant. L'objectif cet algorithme n'est donc pas de trouver la « bonne » réponse. Il est de construire une chaîne de caractère dont on prédit qu'elle apportera un résultat pertinent. De la même manière, la phase de préparation des données est un processus qui sélectionne et normalise certaines informations dont on considère – après un long processus de test et d'essais – qu'elles seront pertinentes dans l'obtention des bons résultats. Enfin, le premier rôle des tests est de quantifier les prédictions. Ils répondent à la question : étant donné les résultats obtenus lors des tests, quelle est la probabilité que la prochaine chaîne de caractères recherchée retourne un résultat pertinent ? Cette approche prédictive implique nécessairement un degré d'incertitude, et donc le développement d'algorithmes flexibles

qui cherchent à minimiser le bruit.

Être conscient de la nature prédictive de ce processus et quantifier la qualité des algorithmes à l'aide de tests permet cependant de prendre de meilleures décisions techniques. La lecture distante et la détection de motifs supposent d'avancer « à l'aveugle », en s'appuyant sur sa connaissance de la structure du texte pour extraire les bonnes informations. Étant donné qu'il est impossible d'être totalement certain que les bonnes données ont été extraites, l'étape suivante – le lancement des requêtes sur l'`API` – doit malléable et s'adapter aux données disponibles. C'est pourquoi le parti pris a été de concevoir un algorithme qui continue de lancer des requêtes en retirant des paramètres tant qu'un identifiant n'a pas été trouvé.

4.2 Un algorithme de détection de motifs pour préparer et structurer les données

Avant de chercher à récupérer un identifiant *Wikidata* via l'`API`, un algorithme se charge de traduire et de structurer les données : à partir d'un nom et de son éventuelle description, un dictionnaire qui contient les informations de manière structurée est construit. Cette étape était initialement censée être une simple extraction d'information : à partir du `name` et du `trait`, un ensemble d'informations étaient mises bout à bout afin de former une chaîne de caractères à rechercher sur l'`API`. Le processus s'est complexifié pour intégrer l'extraction, la traduction et la structuration des données. En construisant un dictionnaire à partir de texte, il est possible de savoir précisément quelles données sont disponibles pour lancer des requêtes ; plusieurs requêtes peuvent alors être lancées sur l'`API` avec différents paramètres, ce qui permet d'augmenter les probabilités d'obtenir un identifiant valide.

4.2.1 Présentation générale

Les formats d'entrée et de sortie

Le but de l'extraction de données permet de transformer la représentation `TEI` visible en 4.10 – représentée sous forme d'un `TSV` pour faciliter la lecture des données – au dictionnaire visible en 4.11. Voici la significtion des différentes clés[8] du format de sortie :

— `fname` : cette clé permet d'accéder au prénom d'une personne. Les données contenues dans cette clé viennent du `name`. `fname` est l'abréviation de « first name ».

8. Une clé de dictionnaire est l'élément à gauche des « : » ; la clé permet d'accéder à la valeur, visible à droite du « : », ce qui permet d'associer des valeurs entre elles, et donc de stocker des objets ou de remplacer une clé présente dans un texte par une valeur, par exemple.

```xml
<item n="271" xml:id="CAT_000327_e271">
    <!-- ... -->
    <name type="author">Turenne (Henri de La Tour d'Auvergne vicomte
    ↪ de)</name>
    <trait>
        <p>illustre maréchal de France, né en 1611, tué en 1675.</p>
    </trait>
    <!-- ... -->
</item>
```

Code source 4.10 – L'entrée `XML-TEI` à partir de laquelle des données sont extraites

— `lanme` : cette clé permet d'accéder au nom de famille de quelqu'un. C'est cette information, extraite du `name`, qui est centrale aux requêtes. `lname` abréviation de « last name »)

— `nobname_sts` : cette clé contient un nom de famille noble. Dans ces cas, un titre de noblesse est présent dans les entrées de catalogue (seuls les titres de noblesse les plus importants sont extraits du dictionnaire, ce qui n'est pas le cas ici). Les informations contenues ici proviennent du `name`. Cette clé est l'abréviation de « nobility name_status »

— `status` : le statut d'une personne, soit son titre de noblesse (ici, le titre « vicomte » n'a pas été extrait car il est rarement présent sur *Wikidata*). Les informations contenues ici proviennent en général du `name` et parfois du `trait`.

— `dates` : les dates de naissance ou de mort d'une personne (seules ces dates sont conservées). Ces informations proviennent du `trait`.

— `function` : la fonction d'une personne, soit, en général, son métier ou son occupation principale. Cette information provient du `trait`.

— `rebuilt` : un booléen indiquant si un prénom a été reconstruit à partir d'initiales ou non.

Comme cela a été dit auparavant, le nom attribué à ces clés n'est pas systématiquement indicateur des valeurs qui y sont associées : si l'entrée de catalogue correspond à une personne, alors les clés correspondent aux informations qu'elles contiennent. Si l'entrée de catalogue ne correspond pas à une personne, ces clés seront également utilisées. Ce qui est important, c'est la hiérarchie d'importance entre les différentes clés : `lname` est la clé centrale et contient presque toujours des informations, `fname` des données secondaires et `date` des dates. Les autres clés sont rarement utilisées si l'entrée de catalogue ne correspond pas à une personne.

```json
{
    "fname": "henri ",
    "lname": "la tour d'auvergne",
    "nobname_sts": "Turenne ",
    "status": "",
    "dates": "1611 1675 ",
    "function": "marshal",
    "rebuilt": False
}
```

Code source 4.11 – La sortie `JSON` correspondante

Présentation de l'algorithme d'extraction d'informations

L'algorithme détaillé ci dessous est présenté sous forme graphique dans la figure 4.3. Cette étape peut être séparée en deux parties différentes : l'extraction d'informations nominatives du `name` et la récupération de données biographiques du `trait`.

L'extraction d'informations du `name` est l'étape plus complexe. La difficulté tient au fait que cette balise peut contenir des informations variées et structurées de façon très différente. Cela demande d'identifier des motifs récurrents et de les repérer dans le texte à différents degrés et à différentes étapes. Dans un premier temps, les prénoms sont systématiquement extraits. Ils sont toujours détectés – même si la balise ne contient le nom d'une personne : cette extraction permet justement d'identifier le type de données contenues dans le `name`. Les prénoms sont repérés à l'aide de plusieurs expressions régulières qui permettent d'identifier des prénoms complets et abrégés, qu'ils soient composés ou non. Cette détection prend en compte les différents types d'abréviations possibles (un prénom composé peut être entièrement abrégé ; à l'inverse, seulement un des prénoms peut être abrégé) et les différentes typographies (séparer les prénoms avec des traits d'union ou non, par exemple). Dans le cas où un prénom serait abrégé, il est si possible reconstruit : des initiales sont remplacées par un nom complet ; ce processus est présenté plus en détail ci-dessous. Cela permet d'augmenter le taux de réussite dans l'alignement avec des identifiants *Wikidata*, mais vient avec plusieurs difficultés techniques, comme nous le verrons. Une fois ce nom extrait, le type d'information contenue dans le `name` doit être identifié : en fonction du type d'information (géographique, historique, nominative...), différents traitements sont mis en place. Cette identification se fait par détection de motifs augmentée par l'usage de tables de conversion [9] et de listes contenant du vocabulaire spécifique. Listes et tables étant classées thématiquement, il est possible, par un processus éliminatoire, d'identifier avec certitude le type d'information contenue dans le `name`. Le traitement du `name` dépend grandement de cette détection : si cette balise contient des éléments géographiques ou historiques, l'extraction d'informations repose en grande

9. Pour un exemple de table de conversion, voir B.1

partie sur les tables de conversion. S'il s'agit en revanche d'un nom de personne, il est alors nécessaire d'identifier les différents types de données nominatives (prénom, nom de famille, nom de famille noble...) pour bien structurer les données. En effet, c'est de cette structure que dépend la bonne construction du dictionnaire, et donc la constitution de requêtes adaptées à l'`API`. Ce processus d'extraction des données s'appuie majoritairement sur une détection de motifs. Le motif déterminant est la présence ou non dans parenthèses dans le nom. Comme nous le verrons, si un nom contient des parenthèses, les informations sont bien plus structurées que s'il n'en contient pas. Les informations peuvent alors être extraites avec une bien plus grande granularité.

Une fois les informations nominatives extraites du `name`, il reste à extraire les données biographiques pertinentes du `trait`. Cette étape, plus simple que la précédente, vaut principalement pour les entrées où c'est l'auteur.ice d'un document qui est mentionné.e dans le `name`. Les seules informations extraites concernent les dates de vie et de mort des personnes, ainsi que son métier. Quelques difficultés techniques subsistent cependant : il faut notamment distinguer une date de naissance/décès d'une autre date, afin de diminuer le bruit ; de plus, il faut réussir à extraire de façon automatique l'occupation principale d'une personne, lorsque plusieurs personnes sont mentionnées (« militaire » et « auteur », par exemple). La résolution de ces deux problèmes repose sur une bonne connaissance du corpus de catalogues et de la structure des `trait`.

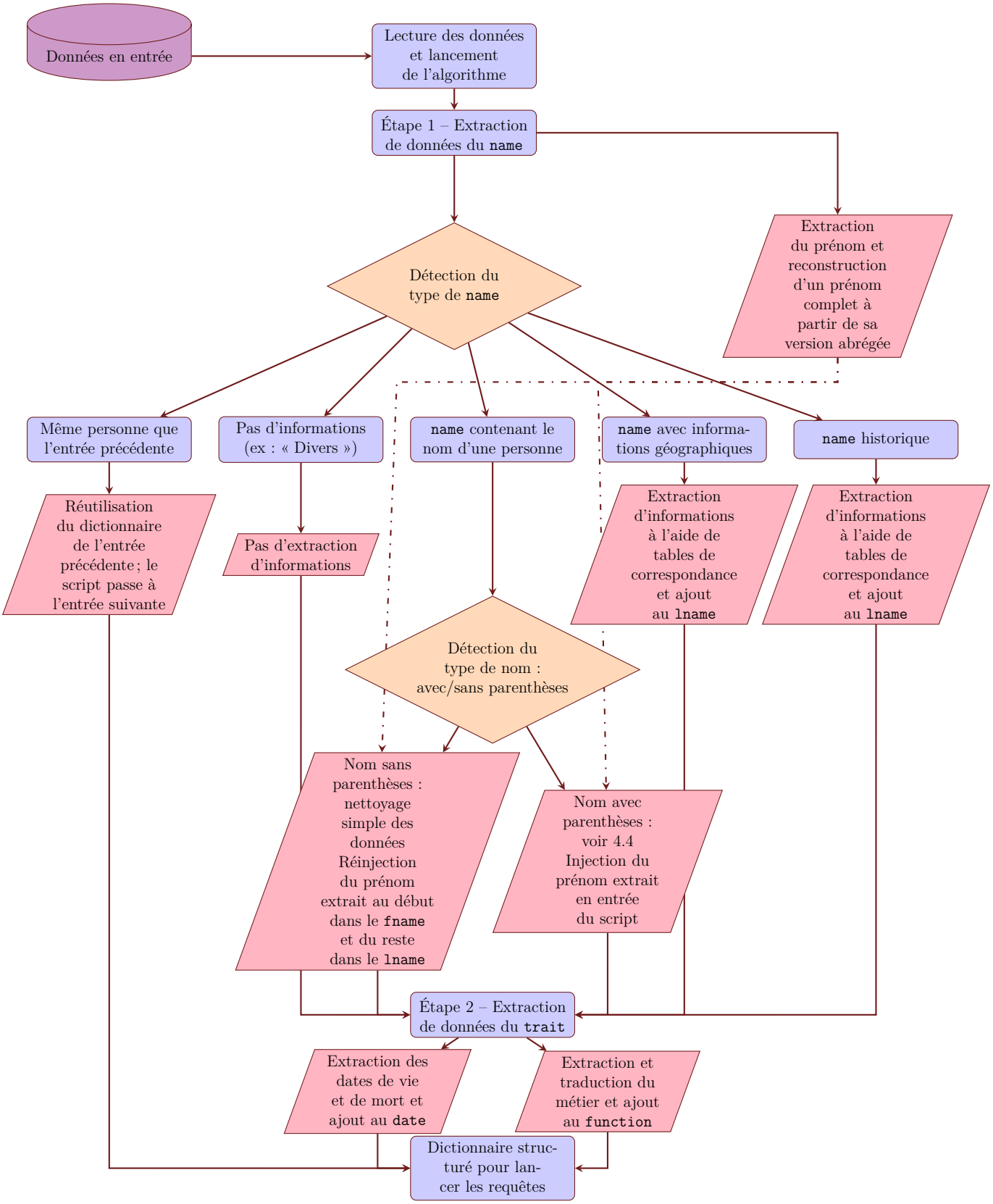


FIGURE 4.3 – Processus d'extraction d'informations du `name` et `item`

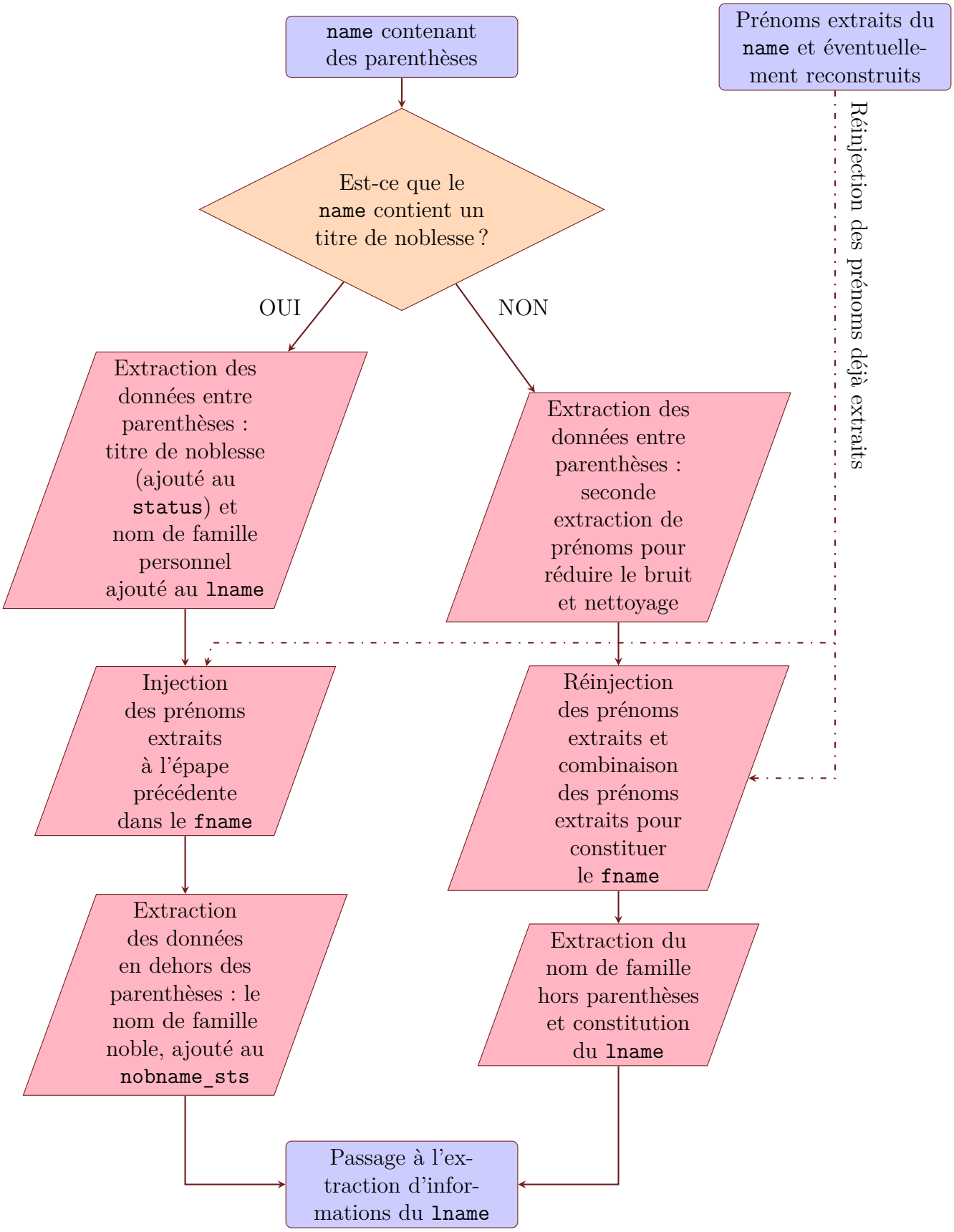


FIGURE 4.4 – Extraction d'informations d'un `name` contenant des parenthèses

4.2.2 Identifier le type de nom

Les éléments `tei:name` contiennent le titre donné à l'item vendu. Si c'est souvent le nom de l'auteur.ice du document, ce n'est pas toujours le cas. L'extraction d'éléments du `name` dépend, comme cela a été dit, de l'identification du « type » de nom. La décision a été prise de classer tous les `name` en cinq catégories.

Les noms génériques : ces éléments ne contiennent pas d'informations précises. Les entités *Wikidata* étant spécifiques plutôt que génériques, il n'est pas certain que les entrées puissent être alignées avec *Wikidata* ; si des entités *Wikidata* correspondent à ces éléments, les informations qu'elles contiennent seront probablement trop génériques pour être utilisables dans un contexte économétrique. Lorsque cela est possible, l'alignement est quand même fait ; c'est le cas par exemple pour les `name` contenant la mention de chartes. Dans ces cas, un dictionnaire vide est retourné et l'alignement avec *Wikidata* n'aura pas lieu. Une exception est faite dans le cas chartes, où un alignement est fait avec l'entité *Wikidata* génériques. Dans cette catégorie se trouve par exemple :

— `<name type="author">DIVERS</name>`

Les noms de type « Le même » ou « La même » ; lorsqu'il y a cette valeur dans le `name`, l'auteur.ice est la même personne que l'auteur.ice de l'entrée de catalogue précédente. Dans ce cas, le dictionnaire de cette entrée est réutilisé. Par exemple :

1. `<name type="author">Le même</name>`.

Les noms géographiques ; ces entrées sont détectées à l'aide de tables de conversion et de listes. Celles-ci sont classées thématiquement : liste d'anciennes colonies françaises (B.4), de départements français du XIX$^{\text{ème}}$ s.(B.2), d'anciennes provinces françaises (B.5) et de pays (B.3). Dans ce cas, un alignement avec une entité *Wikidata* est possible ; cependant, il n'est pas toujours envisageable de retrouver l'entité précise. En effet, un `name` peut contenir une mention d'une donnée géographique, mais également d'autres détails. C'est par exemple le cas dans le quatrième exemple ci-dessous. Il faut alors aligner le `name` avec son équivalent générique sur *Wikidata* (l'exemple ci-dessous, par exemple, a été aligné uniquement avec l'entité « Paris »). Dans cette catégorie se trouvent :

— `<name type="other">AISNE (département de 1')</name>`

— `<name type="author">Bourbonnais. </name>`

— `<name type="author">Paris : Musée royal du Louvre</name>`

— `<name type="author">Garde nationale parisienne en 1792 (brevet de volontaire de la)</name>`

Les noms correspondant à des évènements historiques. Là encore, une table de conversion est utilisée (B.6). Ici, une difficulté apparaît cependant : du fait de la varité des évènements historiques mentionnés dans les entrées de catalogue, il n'est pas possible

d'enregistrer l'ensemble des évènements dans des tables afin de permettre une détection de tous les évènements. Les `name` ont donc été analysés pour extraire les évènements les plus importants. Ensuite, comme pour les termes géographiques, il n'est pas possible de donner aux tables de conversion une granularité suffisamment fine pour contenir toutes les données possibles. Des alignements partiels ont donc été faits : le premier exemple ci-dessous a été aligné avec l'entité *Wikidata* « Révolution française ».

— `<name xmlns="http://www.tei-c.org/ns/1.0" type="other">THÉATRE RÉVOLUTIONNAIRE</name>`

— `<name type="author">COMMUNE DE 1871.</name>`

— `<name type="other">Siège de La Rochelle en 1628</name>`

Les noms de personnes. Ceux-ci ne sont pas simples à traiter : ils peuvent contenir de nombreuses informations : deux noms de famille (usuels et nobles), titres de noblesse, plusieurs prénoms. Ils ont également une structure variée, comme cela apparaît dans les exemples ci-dessous : les noms peuvent être écrits en utilisant des parenthèses ou non ; un prénom peut être écrit en entier, comme dans premier exemple, entièrement abrégé (comme dans l'exemple 3) ou encore partiellement abrégé, ce qui est le cas dans le troisième exemple. Ces différences, qui ne posent pas de problème à un regard humain, sont autant de problèmes techniques. En effet, la détection de motifs fonctionne uniquement sur des critères formels, ou structurels. Il faut donc, en s'appuyant sur la structure des documents, réussir à distinguer un prénom d'un nom de famille, un nom de famille d'un autre, ou encore un nom abrégé d'un nom complet.

— `<name type="author">Humboldt (le baron Alexandre de)</name>`

— `<name type="author">Taccani Tasca (madame la comtesse)</name>`

— `<name type="author">LEGOUVÉ (G. M. J. B.)</name>`

— `<name type="author">LOUIS XVIII</name>`

— `<name type="author">Duras (Emm.-F. de Durfort, duc de)</name>`

Si l'on pose l'extraction d'informations nominatives sur une personne comme étant l'objectif principal, une difficulté apparaît vite : qu'est-ce qui distingue un nom de personne d'un des autres types de noms ? Les éléments `name` sont voués à contenir des noms propres ; chercher à distinguer les noms propres des noms communs n'a donc pas d'intérêt. Cela est d'autant plus que la graphie varie d'un catalogue à l'autre : dans certains, les majuscules sont signifiantes et pourraient permettre de distinguer noms communs et noms propres, tandis que dans d'autres, l'intégralité du `name` est en majuscule. Une détection de motifs à partir de simples critères formels (du type : « Un nom de personne est un ou plusieurs mots commençant par des majuscules ») n'est pas opérante pour ce corpus. Il n'est pas non plus possible de définir un nom positivement, puisqu'il n'existe aucun critère définitoire

pour un nom propre. Enfin, à cette étape, il n'est pas non plus possible de s'appuyer sur l'extraction de prénoms. L'extraction de prénoms se base, comme on le verra, sur de la détection de motifs ; là encore, ce qui est identifié comme un prénom peut tout aussi bien être un nom propre, et il n'existe à ce stade aucune possibilité pour distinguer un nom propre d'un autre. C'est à ce stade qu'a été décidée l'utilisation de tables de conversion et de listes contenant du vocabulaire thématique : en identifiants des références récurrentes à des évènements ou des lieux dans les `name`, il devient possible de définir les noms de personne négativement. En définitive, un nom de personne, c'est donc ce qui n'est pas autre chose et la détection du type de nom fonctionne donc de manière éliminatoire.

L'algorithme mène donc une série de tests, cherchant à détecter si un `name` rentre dans telle ou telle catégorie. Du fait du fonctionnement technique de `python`, une série de tests éliminatoires doit aller du cas le plus particulier au cas le plus générique afin d'éviter les faux positifs : une fois qu'un élément a été détecté comme appartenant à une catégorie, il ne peut plus être réassigné à une autre. C'est pourquoi l'algorithme commence par chercher à classer les éléments dans les catégories où le taux d'erreur est le plus faible, pour ensuite finir par la catégorie la plus générique : celle des noms de personne. : c'est dans ces catégories que le taux d'erreur est le plus faible.

Le script commence donc par chercher à classer un `name` dans les catégories où les informations sont écrites plus ou moins toujours de la même manière. Il cherche d'abord à identifier si le `name` correspond à « Le même » ou « La même ». Dans ce cas, le `name` est le même que celui de l'entrée précédente et c'est ce dictionnaire qui est réutilisé. Ensuite, les entrées génériques sont détectées. Comme elles contiennent toujours des informations écrites de la même manière (« Documents divers »), le taux d'erreur est là encore très faible. Ces entrées génériques ne contiennent pas d'informations spécifiques ; si un `name` appartient à cette catégorie, alors l'algorithme n'extrait pas d'informations.

Ensuite, si un `name` n'appartient ni à l'une ni à l'autre catégorie, alors l'algorithme cherche à classifier un nom en différentes catégories à l'aide de tables de comparaison et de listes contenant du vocabulaire spécifiques. Ces tables et listes sont classées en deux catégories (données historiques et géographiques) afin de définir des traitements spécifiques ; l'algorithme cherche d'abord à identifier des entrées géographiques, puis historiques. En effet, un bien plus grand nombre de tables contenant des données géographiques existe [10], ce qui augmente les possibilités d'un classement correct. Enfin, le script cherche à identifier des informations historiques dans un `name` (B.6). Si une donnée géographique ou historique est repérée, alors équivalents normalisés de cette donnée sont ajoutés au dictionnaire grâce à l'usage de tables de conversions.

Par processus d'élimination, si aucune de ces informations n'a été détectée, alors il n'est plus possible de classer le `name` dans aucune autre catégorie. Il est alors considéré que

10. Nom d'anciennes provinces françaises (B.5), de départements du XIX$^{\text{ème}}$ s. (B.2), d'anciennes colonies (B.4) et de pays (B.3)

le contenu du `name` est le nom d'une personne. L'extraction d'informations se fait ici plus complexe [11], comme nous le verrons : le script traite le nom différemment en fonction de sa structure (présence ou non de parenthèses dans le nom), puis extrait d'éventuels titres de noblesse, noms de famille noble et nom de famille usuel. Enfin, il extrait des prénoms et cherche à reconstruire un prénom complet à partir de son abréviation. Le processus étant éliminatoire, il n'y a bien sûr aucune certitude totale que le contenu du `name` soit bel et bien le nom d'une personne ; il n'est cependant plus possible de mieux catégoriser les éléments. Cela n'est pas non plus extrêmement important, puisque cet algorithme de classification a un rôle fonctionnel et n'impacte pas la manière dont les identifiants sont récupérés depuis l'`API` *Wikidata*. Il permet principalement d'adopter un fonctionnement modulaire en cherchant à détecter des motifs spécifiques dans les `name` en fonction de la catégorie à laquelle ils appartiennent. La détection de motifs étant « aveugle », le traitement qui est fait pour cette catégorie peut être mis à profit de différents types de données : les mêmes motifs peuvent être recherchés et extraits de différents types d'entrées.

À l'issue de cette phase de classification, il est possible de mieux connaître les types de `name` et leur répartition dans le corpus. Les noms de personnes sont très majoritaires dans les `name` : ils représentent 73305 entrées sur un total de 82913, soit 81,52%. C'est pour ce type de données que l'extraction des données a été pensée ; les informations sont donc extraites avec une granularité plus fine qu'avec le reste du corpus. Viennent ensuite les noms de lieux (8693 entrées) et les éléments divers (550). Pour finir se trouvent les évènements historiques (232 entrées) et les éléments vides [12].

4.2.3 Le traitement des noms de personnes

Peut-être y intégrer la subsection suivante ? 4.4 (qui contient une sous-partie de l'algorithme pour plus de lisibilité)

4.2.4 Reconstruire un prénom complet à partir de son abréviation

Souvent, le prénom d'une personne est écrit en abrégé. Partant de ce constat, un algorithme a été construit pour :

— Repérer lorsqu'un prénom est abrégé, en prenant en compte différents types d'abréviations (nom simple ou composé, nom entièrement ou partiellement abrégé) et des possibles fautes dans les catalogues (un point est oublié à la fin d'une abréviation, par exemple).

11. Pour une représentation graphique de cette étape, voir 4.4

12. Ces chiffres proviennent d'un calcul réalisé à partir du script développé pour identifier le type d'entrée. Ils ont été produits afin de réaliser une représentation graphique du type d'entrée, visible en annexes (A.2)

— Reconstruire un prénom complet à partir de son abréviation, ce qui passe pas un algorithme qui cherche à reconstruire le nom en plusieurs étapes pour obtenir le nom le plus complet possible avec un mimimum d'erreurs.

Ici, l'approche est totalement prédictive : il est impossible d'être certain d'obtenir le bon nom complet à partir de son abréviation ; on peut uniquement prédire que le prénom reconstruit sera conforme au vrai prénom (tel qu'il est écrit sur *Wikidata*) et chercher à maximiser cette certitude.

4.2.5 Extraire des informations normalisées à partir d'un nom et de sa description

Cette sous-section détaille l'utilisation de tables de conversion pour traduire et normaliser certaines donnes importantes (dates de vie et mort, titres de noblesse et fonctions).

BIEN PENSER à parler du problème des informations extraites, mais qui ne se rattachent pas à la personne du `name` (eg : femme du comte...) => comment, par détection de motifs, est-ce que l'on règle le problème.

4.3 Extraire des identifiants *Wikidata*

Une fois un dictionnaire de données normalisées produites, un algorithme lance des recherches en plein texte sur l'`API` de *Wikidata* afin de récupérer des identifiants. L'algorithme lance plusieurs requêtes successivement. L'objectif est de récupérer un identifiant en lançant le moins de requêtes, avec le plus de certitude possible.

4.3.1 Présentation générale

Ici est présenté le fonctionnement général de l'algorithme, qui se comporte différemment en fonction du type de données qu'il a à traiter (personne noble ou non, prénom reconstruit ou non...).

4.3.2 Gérer la montée en charge : optimisation et réduction du temps d'exécution

Le script est assez compliqué, repose sur une `API` et traite un grand nombre de données (plus de 82000 entrées). Il prend donc plus d'une dizaine d'heures à s'exécuter et demande des ressources élevées (la première version du script ne fonctionnait plus sur mon ordinateur après avoir traité 5% du jeu de données). L'optimisation nécessaire de l'algorithme est décrite dans cette sous-section.

4.3.3 Évaluation du script : tests, performance et qualité des données extraites de *Wikidata*

Des tests ont été réalisés pour :

— isoler l'impact de chaque paramètre (élément du dictionnaire) dans l'obtention des bons résultats

— évaluer la qualité de l'algorithme final

— mesurer la performance de celui-ci.

Ces tests, et leurs résultats, sont présentés ici.

4.4 Après l'alignement, l'enrichissement : utiliser `SPARQL` pour produire des données structurées

La récupération des identifiants *Wikidata* est la partie la plus complexe dans l'utilisation de *Wikidata* pour enrichir des données. Après une présentation des informations requêtées via `SPARQL`, le processus d'extraction d'informations et de stockage dans un `JSON` est détaillé.

4.4.1 Quelles données rechercher via `SPARQL` ?

Après avoir expliqué pourquoi développer des méthodes d'enrichissement automatique, une seconde question se pose : quelles sont les données à récupérer ? Cette question n'est pas anodine du fait de la diversité du corpus. Le titre donné à un manuscrit dans un catalogue et inscrit dans le `name` est souvent celui d'une personne, mais ce n'est pas toujours le cas. Il arrive également qu'un manuscrit soit nommé d'après un évènement (la Révolution française), un lieu ou une province (Italie, Poitou), ou encore une typologie de document (des chartes).

Un bref retour sur la manière dont fonctionne `SPARQL` permet de mieux comprendre le problème que peut poser la diversité du corpus. `SPARQL` a l'avantage de permettre de récupérer des données propres sur des bases de données en ligne ; cependant, l'information y est organisée de manière très spécifique, ce qui demande de faire des requêtes précises. Ce langage de requêtes a pour but d'interagir avec une base de données au format `RDF`. Avec ce format – dit sémantique – les données sont organisées en « triplets » sujet–prédicat–objet, où :

— le sujet est la ressource principale.

— le prédicat est une propriété du sujet, qui caractérise une relation avec une autre ressource, l'objet.

— l'objet est une ressource secondaire : c'est la valeur d'un prédicat.

Le principe des triplets RDF est mieux exprimé sous forme graphique (4.5) :

FIGURE 4.5 – Exemple de relation sujet – prédicat – object

Deux particularités supplémentaires définissent les formats sémantiques :

— Toutes les « ressources » peuvent être tour à tour sujet ou objet. L'exemple du dessus, par exemple, aurait pu être réécrit sous la forme : « Les porteuses » a été peint par Natalia Gontcharova. Dans ce cas, Natalia Gontcharova est l'objet et *Les porteuses* est le sujet. Par conséquent, une base de données RDF est une base de donnée en graphes ; elle peut être représentée sous la forme d'un réseau de ressources qui entretiennent des relations bilatérales entre elles. Il n'y a pas de hiérarchie entre les informations, contrairement à une base de données XML classique.

— L'ensemble des ressources et des prédicats d'une base de donnée en graphe sont définis et disposent d'un identifiant unique. Les prédicats, plus particulièrement, sont définis selon une ontologie particulière.

Le deuxième point complexifie la définition des données à récupérer via SPARQL : les prédicats sont décrits avec une grande précision ; par conséquent, une information analogue peut être représentée par différents prédicats dans différentes situations. Dans l'ontologie *Wikidata*, la création d'un texte et la création d'une peinture ne correspondent pas au même prédicat. Pour que les données soient utilisables, il faut être très spécifique quant aux informations recherchées.

Les 18899 entités avec lesquelles les entrées de manuscrits ont été alignées peuvent se classer en de nombreuses catégories. Sur *Wikidata*, une entité est une « instance » d'une classe plus large. En suivant la classification de *Wikidata*, les entités appartiennent aux catégories suivantes [13] :

— personnes humaines ; cette catégorie est la plus fréquente (12090 occurrences)

— noms de familles (3180 entités)

— communes françaises (586 occurrences)

— peintures et sculptures (respectivement 520 et 236 entités)

Cette variété peut s'expliquer en partie par le taux d'erreur dans l'alignement avec Wikidata. Cependant, toutes ces « erreurs » ne correspondent pas forcément à des résultats qui ne sont pas pertinents. Par exemple, l'algorithme peut aligner un.e écrivain.e avec un de ses ouvrages, ou une personne avec son portrait. Des résultats erronés peuvent

13. Un graphique présentant l'ensemble de ces catégories se trouve en annexes (A.1).

toujours garder une forme de pertinence. Il est d'autant plus important de construire des requêtes `SPARQL` qui se concentrent pas uniquement sur des personnes. Cependant, il n'est pas possible de faire une requête qui correspondent à toutes les catégories auxquelles appartiennent le corpus. Le choix a donc été fait de se concentrer sur les catégories les plus pertinentes : les personnes, les familles, et les œuvres artistiques et littéraires. Non seulement ces catégories contiennent la grande majorité du corpus (16026 entités), mais ces catégories sont les plus à même de contenir des entités pertinentes. Il a été choisi de ne pas faire de requête spécifique sur les lieux, puisque peu d'informations sont disponibles pour les entités de la catégorie « communes françaises » sur Wikidata.

Un nombre assez conséquent de données ont donc été requêtées avec `SPARQL`, du fait des spécificités des bases de données en graphes, de la variété des entités *Wikidata* auxquelles les manuscrits sont liées, et enfin du fait de la variété du corpus lui même. Ces informations récupérées correspondent aux différentes catégories de *Wikidata*.

— Pour les personnes et les familles, les informations suivantes sont récupérées sur *Wikidata* :
  — Le genre de la personne ;
  — Sa nationalité, afin de voir si l'origine d'une personne influence le prix d'un manuscrit ;
  — Les langues parlées par une personne ; là encore, l'objectif est d'étudier l'impact de l'origine d'un.e auteur.ice sur un prix.
  — Les date de vie et de mort, afin de placer un manuscrit dans une époque et de voir comment son ancienneté et sa contemporanéité en influencent le prix.
  — Le lieu où une personne est née, où elle a vécu et où elle est morte, pour des raisons analogues.
  — La manière dont la personne est morte. Si cette information peut sembler anecdotique à un public contemporain, les catalogues de ventes sont marqués par un goût du sensationnel, et la manière dont une personne est morte est souvent mentionnée, notamment en cas d'exécutions.
  — La religion d'une personne : il peut être intéressant d'étudier si, et comment, ce critère influence l'évolution d'un prix.
  — Les titres de noblesse d'une personne.
  — L'éducation qu'a reçu une personne, afin de mieux situer ses occupations et d'analyser l'impact du niveau et du type d'éducation sur le prix.
  — L'occupation d'une personne, et les fonctions précises qu'elle a occupées : là encore, il est intéressant de situer l'impact de la carrière sur le prix et de voir quelles occupations sont corrélées avec des prix élevés sur le marché des manuscrits.

  - Les prix et distinctions reçus par une personne. À l'aide de ce critère, il est alors possible de chercher à répondre à cette question : la célébrité d'une personne de son vivant impacte-elle le prix de se manuscrits ?
  - Les organisations et institutions dont la personne est membre (Académie française, Franc-maçonnerie...)
  - Le nombre d'œuvres écrites ou réalisées par une personne. Là encore, c'est une tentative de mesurer l'impact ou la célébrité d'une personne : les manuscrits de quelqu'un ayant beaucoup écrit sont ils plus chers que les manuscrits d'une personne ayant peu écrit ?
  - Le nombre de conflits auxquels une personne a participé. Ce critère de recherche permet de quantifier l'importance d'un personnage militaire.
  - Des images, telles que le portrait et la signature.
- Pour les créations littéraires, ce sont des informations bibliographiques qui sont avant tout récupérées ; pour les autres œuvres d'art, des informations analogues sur le contexte de création sont retenues.
  - Le titre de l'œuvre.
  - Son auteur.ice, pour étudier si certain.e.s auteur.ice.s sont susceptibles d'influencer le prix d'un manuscrit.
  - La date de création de l'œuvre, afin de savoir si l'époque d'origine influence le prix. Pour les livres, la date publication est également récupérée.
  - La requête récupère aussi la maison d'édition d'un livre.
  - Les dimensions et matériaux d'une œuvre d'art sont également d'intérêt.
  - Enfin, le genre et le mouvement dans lequel s'inscrivent une œuvre sont d'intérêt : ces informations pourraient permettre de voir si une hiérarchie des du goût influence le prix d'un manuscrit.
- Pour finir, afin de pouvoir éventuellement enrichir nos données avec d'autres sources externes à *Wikidata*, des identifiants uniques ont été récupérés afin de donner accès à d'autres bases de données en ligne : les identifiants VIAF (Fichier d'autorité international virtuel), ISNI (International Standard Name Identifier), de la Bibliothèque nationale de France, de la Bibliothèque du Congrès américain, ainsi que les identifiants IDRef. Certaines insitutions, comme la BnF, rendent leurs données accessibles via `SPARQL` ; la récupération de ces identifiants faciliterait grandement les enrichissements ultérieurs depuis d'autres sources de données.

Comme on l'a dit, l'objectif principal de l'alignement avec *Wikidata* est de produire des données pour calculer des régressions linéaires, ce qui permettrait d'étudier les déterminants du prix d'un manuscrit sur le marché du XIX$^{\text{ème}}$ s.. Cette récupération

d'informations en masse ouvre d'autres possibilités. Entre autres, de nombreuses données géographiques ont sont récupérées (lieu de naissance, de mort, d'enterrement). Il est ensuite possible de récupérer les coordonnées de ces lieux, afin de construire une cartographie des auteur.ice de manuscrits circulant sur le marché parisien du XIX$^{\text{ème}}$ s. parisien. Cette possibilité n'est pas anodine, puisqu'elle permettrait de mettre en relation la « parisianité » avec la construction du canon littéraire à Paris. Il serait également possible d'étudier la circulation des productions culturelles, et leur rayon d'influence. En croisant les données géoréférencées avec des données chronologiques (dates de naissance et de mort...), ces questions peuvent également être étudiées de façon historique : comment l'influence de l'origine géographique sur la réception d'une œuvre évolue au fil des siècles ? Répondre à ces questions n'a pas été possible dans le cadre de mon stage ; cependant, grâce à l'enrichissement de données via `SPARQL`, il de telles études deviennent possibles, et les données pour mener ces analyses sont au moins en partie déjà disponibles. Produire des informations normalisées et exploitables pour la recherche implique donc de produire des données réutilisables, qui doivent être réutilisables avec d'autres problématiques de recherches.

4.4.2 Présentation générale

Comme pour les autres étapes, on présente ici, à l'aide d'un schéma, la structure générale de l'algorithme de requêtes.

4.4.3 Développer un comportement uniforme pour produire des données exploitables à partir un corpus hétérogène

Ici est détaillée

— la requête `SPARQL` lancée (subdivisée en plusieurs petites requêtes).

— le format de sortie produit à partir des données renvoyées par `SPARQL`

4.4.4 Minimiser la perte : optimisation et gestion des erreurs

Comme des quantités massives de requêtes sont lancées, et que de très nombreuses informations sont demandées, des erreurs peuvent avoir lieu, et notamment des erreurs de *timeout* (le temps d'exécution dépasse la durée autorisée). La gestion de ces erreurs est décrite ici.

4.4.5 Lier la `TEI` aux données nouvellement produites

Cette courte section détaille la mise à jour des fichiers `TEI` avec les identifiants *Wikidata*, ce qui permet de faire le lien entre les entrées de catalogues et les données

issues de *Wikidata*.

4.5 Des données à la monnaie : premiers résultats de l'étude

Sous réserve que l'étude des régressions linéaires ait été fait à temps (ce qui n'est pas garanti), j'aimerais ici présentés les premiers résultats sur les facteurs de l'évolution des prix.

Troisième partie

Après la TEI : l'application web *Katabase*, interface de diffusion des données

Chapitre 5

Design d'interfaces dans un projet d'humanités numériques : l'application web *Katabase*

Ce chapitre s'intéresse aux relations entre *web design*, données textuelles et humanités numériques, à partir de l'exemple du site web développé pour le projet *Katabase*.

5.1 Le design d'interfaces : une reconfiguration des méthodes de recherche et une transformation du corpus

Cette section s'intéresse aux nouveautés apportées par le design d'interfaces dans les humanités numériques. On s'intéresse à la manière dont le design d'interfaces (et le design de façon générale) transforme les méthodes de recherche « habituelles », mais aussi une transformation du rapport aux documents.

5.1.1 Le design comme inversion des méthodes

Avec les humanités numériques, les questions de design et de structuration deviennent centrales, depuis la conception de schémas `TEI` (qui demandent de mettre en forme un document pré-existant) et d'ontologies jusqu'au développement d'interfaces et de sites web. Parmi ces questions « formelles », le design d'interfaces occupe cependant une place particulière. En effet, dans la plupart des aspects des humanités numériques, le rapport entre questions techniques et scientifiques est clairement établi ; la question scientifique préexiste, et la technique sert surtout à répondre à cette question (comme cela a été le cas jusqu'à dans la « pipeline » jusqu'ici). Cette hiérarchie entre théorie et

pratique reste somme toute assez traditionnelle et correspond aux méthodes scientifiques établies.

Avec le design d'interfaces, ce rapport établi s'inverse. En effet, le design ne cherche pas à répondre à une question. Tout au plus, il répond à un cahier des charges (il faut, à minima, permettre de diffuser des données de façon lisible par des êtres humains). C'est avec la pratique du design que naissent les problématiques, parmi lesquelles :

— Comment organiser les différentes parties d'une page pour que celle ci soit lisible ?

— Comment organiser la relation entre les pages pour qu'un site web soit facilement navigable ?

— De quelle manière l'apparence d'un site détermine la réception des contenus ?

— En quoi le design d'un site web construit ou bouscule des habitudes et des formes d'utilisation chez ses utilisateur.ice ?

Toutes les questions posées par le design n'attendent pas nécessairement de réponse. Cependant, force est de constater que ce domaine appelle à une nouvelle approche pour des chercheur.euse.s et ingénieur.e.s issu.e.s des humanités ; ces questions visuelles amènent à une approche semblable à celle de la recherche-création et demandent de développer un nouveau rapport à la technique.

5.1.2 Interface et document

En plus de perturber nos méthodes, la conception d'interfaces influence la perception des documents. Dans le cas du projet *Katabase*, le site web opère une médiation, il implique de une « scénographie » autour des catalogues de vente. Ceux-ci et les manuscrits qui y ont décrits sont intégrés à des pages, inclus dans un parcours, accessibles depuis différents points d'entrée. En plus de cette scénographie, les catalogues sont littéralement traduits, depuis la `TEI` vers le format `HTML`. Là où la `TEI` est un format de balisage sémantique (c'est la signification des éléments guide l'encodage), le `HTML` est pensé pour un balisage formel (le texte est balisé en fonction de la forme que l'on souhaite obtenir). Cela implique une perte d'information (les métadonnées du `teiHeader`) et l'éloignement d'une approche philologique du texte. Enfin, le site internet marque un éloignement intellectuel avec les documents : le catalogue n'y est plus l'unité intellectuelle dominante, alors qu'il restait l'un des critères structurants des fichiers `TEI` (un fichier représentant un catalogue). Sur le site web, on peut accéder directement aux éléments vendus, sans avoir à passer par les catalogues. Dans le contexte d'un projet issu de la littérature, toutes ces opérations ne sont pas neutres et méritent d'être explicitées. Pour mieux identifier ce que ces transformations impliquent, il peut être intéressant de revenir à la « roue » de Sahle [1].

1. P. Sahle, « Digital modelling. Modelling the digital edition »…, p. 11.

5.2 La conception d'interfaces, un problème pour les humanités numériques ?

Cette section s'intéresse aux rôle des interfaces en humanités numériques.

5.2.1 Pour une approche pragmatique du design d'interfaces dans un contexte d'humanités numériques

Le design graphique demande des compétences spécifiques qui ne font pas directement partie des cursus d'humanités numériques. Il ne sert pas seulement à faire des sites qui soient « beaux », il joue un rôle essentiel en encadrant la réception des contenus présentés. Cependant, les approches plus « élaborées » de design d'interfaces demandent des financements et des techniques qui sont souvent hors de portée d'un projet universitaire. Des approches plus « critiques » du design ont également été développées dans les humanités numériques [2]. Ces approches ont tendance à être difficiles à mettre en œuvre ; leur portée critique peut aller à l'encontre de l'utilité des interfaces, en faisant de l'interface l'objet principal d'intérêt, aux dépends des contenus présentés.

À l'opposé de ces approches, ce qui est défendu dans le cadre du projet *MSS / Katabase* est une approche à la fois informée et pragmatique du *web design*. Informée, car être conscient des enjeux du design permet un meilleur positionnement en tant qu'ingénieur.e, et donc une présentation des contenus plus intéressante. Pragmatique, parce que les solutions qui sont présentées sont des solutions techniquement réalisables dans le cadre d'un projet universitaire. C'est ici qu'est présentée la charte graphique développée pour l'application web *Katabase*.

5.2.2 Rejeter les interfaces ?

Après avoir parlé de l'intérêt des interfaces et présenté l'approche suivie au sein du projet *MSS / Katabase*, cette partie s'attache à développer une critique des interfaces. À partir d'une approche historique des interfaces graphiques, des contextes dans lesquelles elles se sont développées, nous revenons sur les concepts centraux à leur développement que sont la notion d'utilisateur et de design d'expérience. Il ne s'agit pas de remettre en cause l'utilisation d'interfaces, mais de défendre une approche critique et consciente de l'impact que la standardisation des « expériences utilisateur » sur internet peuvent avoir sur la diffusion des connaissances.

2. Johanna Drucker, *Visualisation : l'interprétation modélisante*, Paris, 2020 (Esthétique des données, 03).

Chapitre 6

Donner à voir un corpus textuel

Ce chapitre s'intéresse aux visualisations développées pour l'application web *Katabase*.

6.1 Visualisation, design et sciences : des relations complexes

Ici, on s'intéresse à la place qu'occupe la visualisation de données dans la recherche scientifique. Le rapport entre les sciences et la visualisation est loin d'être simple et unidirectionnel : cette dernière n'est pas juste un outil, une méthode utilisée dans la recherche scientifique pour des raisons pratiques. Il est plus intéressant de penser la visualisation (et donc le design) et les sciences comme des domaines en interaction, qui s'influencent mutuellement. De la même manière que l'écriture implique des manières de penser particulières [1] en donnant au discours une existence spatiale (le texte est répandu sur une page et des renvois peuvent être fait d'un endroit de la page à un autre), la visualisation implique ses propres manières de penser et influence donc la recherche. Dans notre cas par exemple, produire des visualisations implique de s'intéresser à des informations quantifiables ; cela encourage donc une approche statistique du corpus. À l'inverse, la recherche scientifique ne fait pas qu'« utiliser » le design. Certaines pratiques sont favorisées et deviennent force d'autorité dans des disciplines scientifiques. Se créent alors des « cultures visuelles » [2] propres à ces disciplines, avec leurs traditions et motifs.

1. Anthony Masure, *Design et humanités numériques*, Paris, 2017 (Esthétique des données, 01), p. 111-116.

2. Klaus Hentschel, *Visual cultures in science and technology. A Comparative History*, Oxford, 2014, p. 14.

6.1.1 L'utilisation de supports visuels dans les sciences : une longue histoire

Ici, on retrace une histoire de l'utilisation du visuel dans les sciences (au sens large : sciences « dures » et sciences humaines), à partir (entres-autres) du travail de Anne-Lyse Renon [3] et de K. Hentschel [4].

6.1.2 Une vision objective ? Visualisation et prétention à l'objectivité

Cette partie fait un retour sur la manière dont le visuel et la production de graphiques ont été utilisés comme arguments d'autorité, afin de montrer des faits de façon « objective » [5].

6.1.3 La tendance visuelle des humanités numériques

Pour finir, on fait un bref retour sur la manière dont les humanités numériques « intensifient » la tendance à la visualisation, ou complexifient le rapport entre sciences et visualisation, pour deux raisons. En premier lieu, les humanités numériques viennent avec le développement de nouveaux outils. Ensuite, les humanités numériques marquent un retour à une approche quantitative et graphique dans les sciences humaines – approche qui trouve ses sources, entre-autres, dans l'École des Annales et sa collaboration avec le Laboratoire de graphique de Jacques Bertin [6], ainsi que dans le structuralisme, où les « structures » trouvent leur meilleure représentation sous forme graphique. Cette tendance visuelle des humanités numériques n'est pas sans poser problème, puisque les visualisations sont développées par des personnes qui n'ont pas nécessairement de formation en graphisme. Une approche pragmatique de la visualisation a tendance à primer (les graphiques servent à prouver quelque chose), aux dépens d'une approche critique (les représentations graphiques sont des interprétations, où les données sont signifiantes, mais où les formes et les méthodes de représentation importent aussi).

---

3. Anne-Lyse Renon, *Design et esthétique dans les pratiques de la science*, Thèse de doctorat, Paris, École des Hautes Études en Sciences Sociales, Institut Marcel Mauss, 2016, URL : `https://www.academia.edu/36754513/Design_et_esth%C3%A9tique_dans_les_pratiques_de_la_science` (visité le 08/06/2022), p. 47-88.

4. K. Hentschel, *Visual cultures in science and technology. A Comparative History…*

5. A.L. Renon, « "Design graphique" et "objectivité", la question des méta-altas », dans *Voir l'architecture. Contribution du design à la construction des savoirs*, dir. Annick Lantenois et Gilles Rouffineau, Paris, Grenoble, 2015, p. 71-81.

6. Olivier Orain, « Le Laboratoire de cartographie dans le contexte de développement des sciences sociales et humaines, des années 1950 aux années 1970 », dans *Design graphique, recherche et patrimoine des sciences sociales. Le Laboratoire de graphique de Jacques Bertin*, Pierrefitte-sur-Seine : Archives nationales, 2021, URL : `https://dai.ly/x85jbir` (visité le 14/07/2022).

6.2 Interpréter le corpus de manuscrits

Ce chapitre s'intéresse à la manière dont le corpus de catalogues de vente de *Katabase* a été traduit en graphiques et à la manière dont ces représentations permettent un nouveau regard sur le corpus.

6.2.1 La visualisation comme objet de connaissance

Ici, on développe une analyse du corpus de catalogues et des manuscrits qui y sont décrits à partir des visualisations produites. Par leur capacité à traduire les informations sous des formes synthétiques [7], les visualisations sont des objets de connaissance qui permettent de comprendre le corpus traité.

6.2.2 La visualisation comme interprétation

Les représentations graphiques ne font pas que montrer des phénomènes. Leur rôle est moins analytique que démonstratif : elles ne révèlent pas une information qui serait cachée dans les données, mais interprètent celles-ci conformément à une problématique de recherche [8]. Représenter un jeu de données, c'est donc le lire, l'interpréter en fonction de certaines questions scientifiques. Ce processus interprétatif est donc partiel (on ne dit pas tout ce qui est dans un jeu de données, mais seulement ce qui est pertinent dans un certain contexte) ; il est aussi influencé par les propriétés graphiques des visualisations. Cette sous-section s'intéresse donc, à partir d'exemples concrets, à la manière dont les propriétés graphiques (choix de formes et de couleurs) ainsi que le pré-traitement des données et d'autres décisions techniques (représentation des prix en francs courants ou constants) influencent la lecture et la perception du corpus.

7. K. Hentschel, *Visual cultures in science and technology. A Comparative History…*, p. 36.
8. J. Drucker, *Visualisation : l'interprétation modélisante…*, p. 78.

Bibliographie

Projet *MSS / Katabase*

Corbières (Caroline), *Du catalogue au fichier TEI. Création d'un workflow pour encoder automatiquement en XML-TEI des catalogues d'exposition.* Mémoire pour le diplôme de master "Technologies numériques appliquées à l'histoire", Paris, École nationale des Chartes, 2020, URL : `https://github.com/carolinecorbieres/Memoire_TNAH` (visité le 13/06/2022).

Gabay (Simon), Rondeau du Noyer (Lucie) et Khemakhem (Mohamed), « Selling autograph manuscripts in 19th c. Paris : digitising the Revue des Autographes », dans *IX Convegno AIUCD*, Milan, 2020, URL : `https://hal.archives-ouvertes.fr/hal-02388407` (visité le 13/06/2022).

Gabay (Simon), Rondeau du Noyer (Lucie), Gille Levenson (Matthias), Petkovic (Ljudmila) et Bartz (Alexandre), « Quantifying the Unknown : How many manuscripts of the marquise de Sévigné still exist ? », dans *Digital Humanities DH2020*, Ottawa, 2020, URL : `https://hal.archives-ouvertes.fr/hal-02898929/` (visité le 13/06/2022).

Janès (Juliette), *Du catalogue papier au numérique. Une chaîne de traitement ouverte pour l'extraction d'information issue de documents structurés*, Mémoire pour le diplôme de master "Technologies numériques appliquées à l'histoire", Paris, École nationale des Chartes, 2021, URL : `https://github.com/Juliettejns/Memoire_TNAH` (visité le 13/06/2022).

Khemakhem (Mohamed), Romary (Laurent), Gabay (Simon), Bohbot (Hervé), Frontini (Francesca) et Luxardo (Giancarlo), « Automatically Encoding Encyclopedic-like Resources in TEI », dans *The annual TEI Conference and Members Meeting*, Tokyo, 2018, URL : `https://hal.archives-ouvertes.fr/hal-01819505` (visité le 13/06/2022).

— « Information Extraction Workflow for Digitised Entry-based Documents. » Dans *DARIAH Annual event 2020*, Zagreb, 2020, URL : `https://hal.archives-ouvertes.fr/hal-02508549` (visité le 13/06/1997).

Rondeau du Noyer (Lucie), *Encoder automatiquement des catalogues en XML-TEI. Principes, évaluation et application à la revue des autographes de la librairie Cha-*

ravay, Mémoire pour le diplôme de master "Technologies numériques appliquées à l'histoire", Paris, École nationale des Chartes, 2019, URL : https://github.com/lairaines/M2TNAH (visité le 13/06/2022).

Rondeau du Noyer (Lucie), Gabay (Simon), Khemakhem (Mohamed) et Romary (Laurent), « Scaling up Automatic Structuring of Manuscript Sales Catalogues », dans *TEI 2019 : What is text, really ? TEI and beyond*, Graz, 2019, URL : https://hal.archives-ouvertes.fr/hal-02272962 (visité le 13/06/2022).

Édition numérique, traitement automatisé et analyse de texte

Bleeker (Elli), Haentjens Dekker (Ronald) et Buitendijk (Bram), « Texts as Hypergraphs : An Intuitive Representation of Interpretations of Text », *Journal of the Text Encoding Initiative*, 14 (2021), DOI : https://doi.org/10.4000/jtei.3919.

Burnard (Lou), « What is TEI conformance, and why should you care ? », *Journal of the Text Encoding Initiative*, 12 (2019), DOI : https://doi.org/10.4000/jtei.1777.

Burnard (Lou), Schöch (Christian) et Odebrecht (Carolin), « In search of comity : TEI for distant reading », *Journal of the Text Encoding Initiative*, 14 (2021), DOI : https://doi.org/10.4000/jtei.3500.

Christensen (Kelly), Gabay (Simon), Pinche (Ariane) et Camps (Jean-Baptiste), « SegmOnto – A Controlled Vocabulary to Describe Historical Textual Sources », dans *Documents anciens et reconnaissance automatique des écritures manuscrites*, Paris : École nationale des Chartes, 2022.

Expression régulière, Wikipedia. L'encyclopédie libre, 2022, URL : https://fr.wikipedia.org/wiki/Expression_r%C3%A9guli%C3%A8re (visité le 10/07/2022).

Gabay (Simon), Camps (Jean-Baptiste), Pinche (Ariane) et Jahan (Claire), « SegmOnto : common vocabulary and practices for analysing the layout of manuscripts (and more) », dans *1st International Workshop on Computational Paleography (IWCP@ICDAR 2021)*, Lausanne, 2021, URL : https://hal.archives-ouvertes.fr/hal-03336528 (visité le 29/06/2022).

Sahle (Patrick), « Digital modelling. Modelling the digital edition », dans *Medieval and modern manuscript studies in the digital age*, London/Cambridge, 2016, URL : https://dixit.uni-koeln.de/wp-content/uploads/2015/04/Camp1-Patrick_Sahle_-_Digital_Modelling.pdf.

— « What is a scholarly digital edition ? », dans *Digital scholarly editing. Theories and practices*, dir. Matthew James Driscoll et Elena Pierazzo, Cambridge, 2016, URL : http://books.openedition.org/obp/3397 (visité le 10/07/2022).

Stokes (Peter A.), Kiessling (Benjamin), Stökl Ben Ezra (Daniel), Tissot (Robin) et Gargem (Hassane), « The eScriptorium VRE for Manuscript Cultures », *Classics@ Journal*, 18 (2021), url : https://classics-at.chs.harvard.edu/classics18-stokes-kiessling-stokl-ben-ezra-tissot-gargem/ (visité le 14/07/2022).

Technologies du web

Covington (Paul), Adams (Jay) et Sargin (Erme), « Deep Neural Networks for YouTube Recommendations », dans *Proceedings of the 10th ACM Conference on Recommender Systems*, New York, 2016, url : https://research.google/pubs/pub45530/ (visité le 10/06/2022).

Harris (Steve), Seaborne (Andy) et Prud'hommeaux (Eric), *SPARQL 1.1 Query Language. W3C Recommendation 21 March 2013*, W3C, 2013, url : http://www.w3.org/TR/2013/REC-sparql11-query-20130321/ (visité le 10/06/2022).

Interface de programmation, Wikipedia. L'encyclopédie libre, 2022, url : https://fr.wikipedia.org/wiki/Interface_de_programmation (visité le 27/07/2022).

Mikolov (Tomas), Chen (Kai), Corrado (Greg) et Dean (Jeffrey), « Efficient Estimation of Word Representations in Vector Space » (, 2013), Publisher : arXiv Version Number : 3, doi : 10.48550/ARXIV.1301.3781.

Moteur de recherche, Wikipedia. L'encyclopédie libre, 2022, url : https://fr.wikipedia.org/wiki/Moteur_de_recherche (visité le 17/07/2022).

Visualisation et design d'interfaces

Agamben (Giorgio), *Qu'est-ce qu'un dispositif ?*, Stanford, 2006 (Rivages Poche / Petite Bibliothèque).

Albouy (Ségolène), *Médiation des données de la recherche. Élaboration d'une plateforme en ligne pour une base de tables astronomiques anciennes.* Mémoire pour le diplôme de master "Technologies numériques appliquées à l'histoire", Paris, École nationale des Chartes, 2019, url : https://github.com/Segolene-Albouy/Memoire-TNAH2019 (visité le 13/06/2022).

Drucker (Johanna), *Visualisation : l'interprétation modélisante*, Paris, 2020 (Esthétique des données, 03).

Hentschel (Klaus), *Visual cultures in science and technology. A Comparative History*, Oxford, 2014.

Masure (Anthony), *Design et humanités numériques*, Paris, 2017 (Esthétique des données, 01).

Orain (Olivier), « Le Laboratoire de cartographie dans le contexte de développement des sciences sociales et humaines, des années 1950 aux années 1970 », dans *Design graphique, recherche et patrimoine des sciences sociales. Le Laboratoire de graphique de Jacques Bertin*, Pierrefitte-sur-Seine : Archives nationales, 2021, URL : https://dai.ly/x85jbir (visité le 14/07/2022).

Renon (Anne-Lyse), « "Design graphique" et "objectivité", la question des méta-altas », dans *Voir l'architecture. Contribution du design à la construction des savoirs*, dir. Annick Lantenois et Gilles Rouffineau, Paris, Grenoble, 2015, p. 71-81.

— *Design et esthétique dans les pratiques de la science*, Thèse de doctorat, Paris, École des Hautes Études en Sciences Sociales, Institut Marcel Mauss, 2016, URL : https://www.academia.edu/36754513/Design_et_esth%C3%A9tique_dans_les_pratiques_de_la_science (visité le 08/06/2022).

Marché de l'art, économétrie et statistiques

F-score, Wikipedia. L'encyclopédie libre, 2022, URL : https://en.wikipedia.org/wiki/F-score (visité le 12/06/2022).

Maupeou (Félicie Faizand de) et Saint-Raymond (Léa), « Les "marchands de tableaux" dans le Bottin du commerce : une approche globale du marché de l'art à Paris entre 1815 et 1955 », *Artl@s Bulletin*, 2 (2013), URL : https://hal.archives-ouvertes.fr/hal-02986371 (visité le 20/06/1997).

Piketty (Thomas), *Les hauts revenus en France au XXe siècle : inégalités et redistributions, 1901-1998*, Paris, 2001.

Précision et rappel, Wikipedia. L'encyclopédie libre, 2022, URL : https://fr.wikipedia.org/wiki/Pr%C3%A9cision_et_rappel (visité le 13/06/2022).

Régression linéaire, Wikipedia. L'encyclopédie libre, 2022, URL : https://fr.wikipedia.org/wiki/R%C3%A9gression_lin%C3%A9aire (visité le 10/07/2022).

Saint-Raymond (Léa), « Revisiting Harrison and Cynthia White's Academic vs. Dealer-Critic System », *Arts*, 8–3 (2019), DOI : https://doi.org/10.3390/arts8030096.

Glossaire

API Une interface de programmation d'application (*Application Programming Interface* en anglais) est un protocole qui pemret à un programme de communiquer avec un autre programme. Ce protocole documenté, correspond à un ensemble d'opérations permettant à un programme « consommateur » d'utiliser des fonctionnalités d'un programme « fournisseur », comme par exemple de récupérer ou d'envoyer des données au fournisseur.. 14

Dictionnaire Un dictionnaire est un format de données structuré en python qui associe à une donnée – dite clé – une ou plusieurs données nommées valeurs.. 18, 29, 32, 37

Expression régulière Une expression régulière, ou expression rationnelle est une chaîne de caractère, écrite selon une syntaxe précise, qui permet de détecter des motifs dans du texte[9]. Les expressions régulières s'appuient sur la classification des caractères en classes (minuscules, majuscules, chiffres, espaces), sur des structures alternatives (un caractère ou un autre) et exclusives (un caractère n'ayant pas certaines propriétés) pour repérer des motifs. Par exemple, « 2022 » correspond au motif : `\d4`, soit « quatre chiffres à la suite ». Une adresse mail est également un motif : `[^(@|\s)]+@[^(@|\s)]+`, soit « plusieurs caractères qui ne sont ni un espace une arobase, suivi d'une arobase, suivi de plusieurs caractères qui n'est ni un espace ni une arobase ».. 14, 18, 28, 31

Python Python est un language de programmation impérative (c'est-à-dire, basé sur la production et la transformation de données par une suite d'instructions) créé en 1991. Il est particulièrement utilisé dans le domaine du traitement de données et des humanités numériques. C'est le langage le plus utilisé dans le projet *MSS / Katabase*. 37

Score F1 Le score F1, ou *F-score*, est la moyenne harmonique de la précision (vrais positifs par rapport au total de résultats obtenus) et du rappel (nombre de résultats

9. *Expression régulière*, Wikipedia. L'encyclopédie libre, 2022, URL : `https://fr.wikipedia.org/wiki/Expression_r%C3%A9guli%C3%A8re` (visité le 10/07/2022).

positifs par rapport au total de résultats positifs).[10]. Le score F1 a l'avantage de prendre en compte les vrais et les faux positifs. Ce score, dont la valeur est contenue entre 0 et 1, permet de mesurer l'exactitude d'un algorithme d'apprentissage machine, ou d'un moteur de recherche.. 14, 19

SPARQL SPARQL est un language de requête qui permet d'interagir avec une base de données au format RDF. « SPARQL exprime des requêtes sur des sources de données diverses [...]. SPARQL rend possible la requête de données en graphes [...], avec des données conjointes et disjointes. SPARQL permet également l'agrégation de données, les sous-requêtes, la négation, la création de données à l'aide d'expression, le test de données et la contrainte des requêtes. Les résultats de requêtes sparql peuvent être des jeux de résultats ou des graphes RDF. »[11]. 13, 14, 18–20, 39–41, 43, 44, 66

Table de conversion Une table de conversion est, tout simplement, un dictionnaire qui contient en clés un certain nombre d'informations telles qu'elles figurent dans le texte et, en valeurs, une version normalisée de cette information. Cela permet de détectier des motifs à extraire autant de normaliser les informations.. 18, 28, 31

10. *Précision et rappel*, Wikipedia. L'encyclopédie libre, 2022, URL : `https://fr.wikipedia.org/wiki/Pr%C3%A9cision_et_rappel` (visité le 13/06/2022) ; *F-score*, Wikipedia. L'encyclopédie libre, 2022, URL : `https://en.wikipedia.org/wiki/F-score` (visité le 12/06/2022).

11. « SPARQL can be used to express queries across diverse data sources [...]. SPARQL contains capabilities for querying [...] graph patterns along with their conjunctions and disjunctions. SPARQL also supports aggregation, subqueries, negation, creating values by expressions, extensible value testing, and constraining queries by source RDF graph. The results of SPARQL queries can be result sets or RDF graphs. »Steve Harris, Andy Seaborne et Eric Prud'hommeaux, *SPARQL 1.1 Query Language. W3C Recommendation 21 March 2013*, W3C, 2013, URL : `http://www.w3.org/TR/2013/REC-sparql11-query-20130321/` (visité le 10/06/2022) (traduction de l'auteur).

Acronymes

API Application Programming Interface. 14, 19, 21, 23, 24, 26, 29, 32, 39, *Glossaire :* API

Table des figures

Table des matières

Annexe A

Graphiques

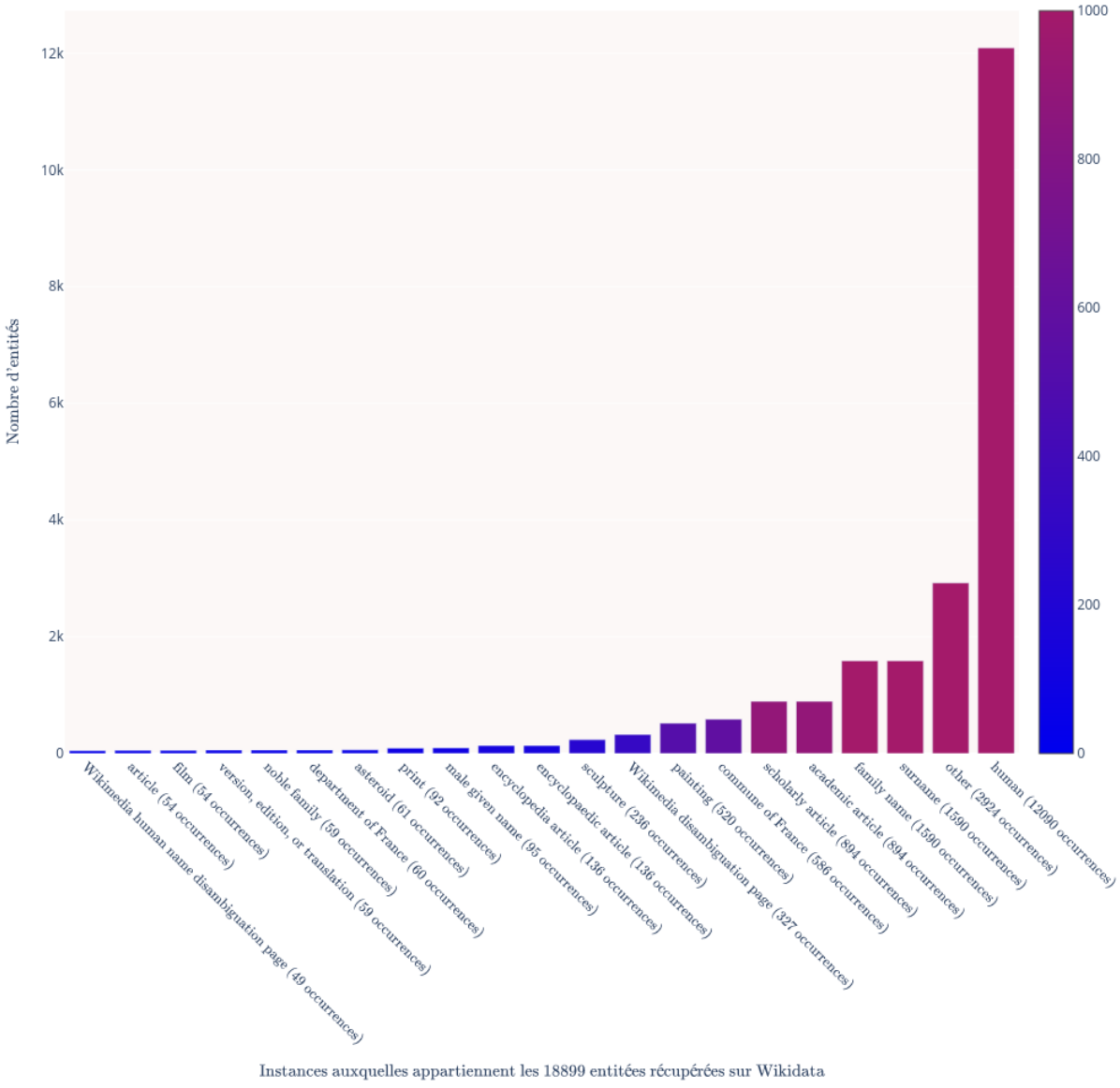


FIGURE A.1 –
Occurrences des différentes catégories auxquelles appartiennent les entités *Wikidata* liées avec les entrées de catalogues

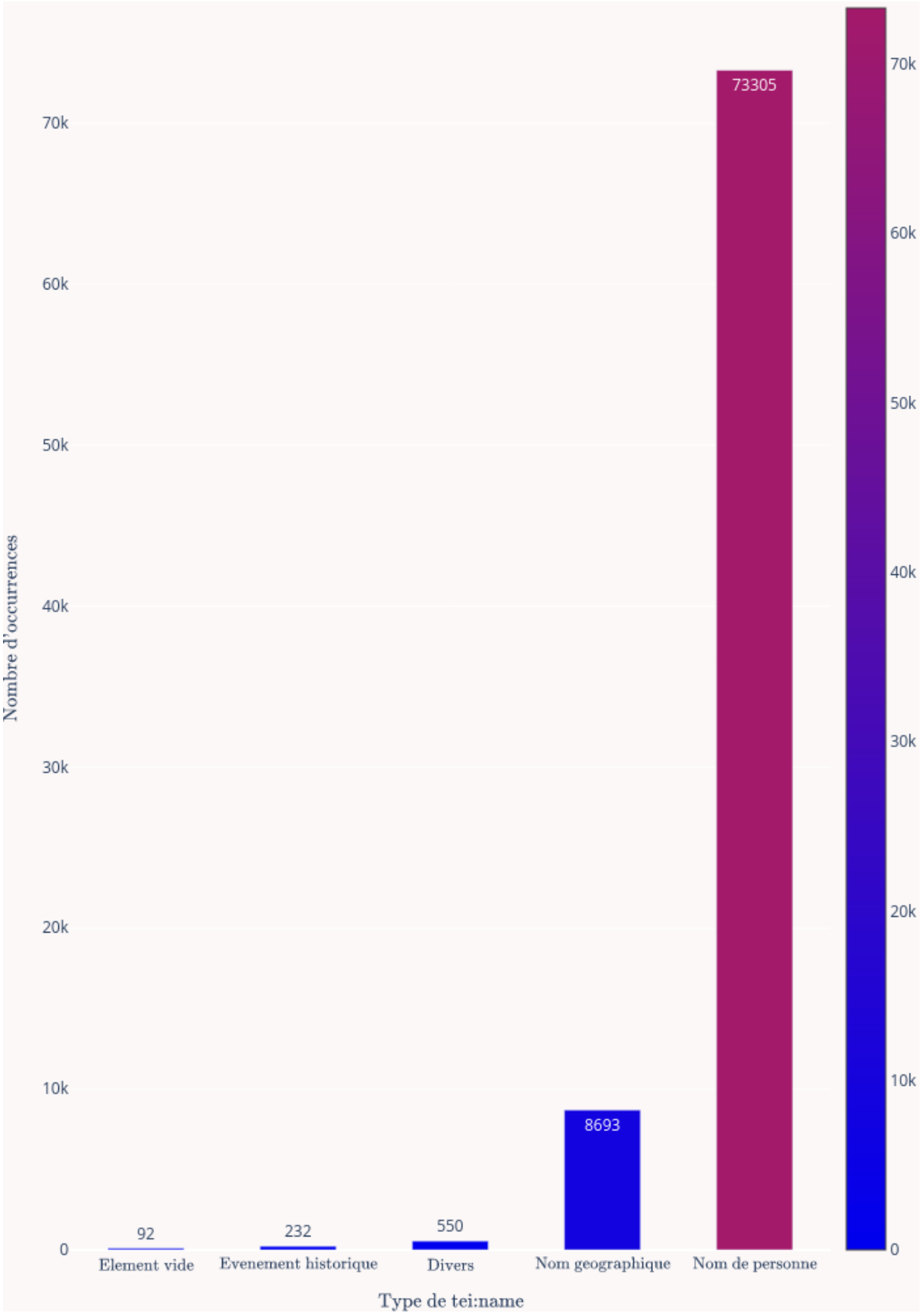


FIGURE A.2 – Répartition des différents types de `name`

Annexe B

Code source et données encodées

```
functions = {
    "général": "general",
    "maréchal": "marshal",
    "lieutenant": "military",
    "officier": "military",
    "colonel": "military",
    "lieutenant-colonel": "military",
    "commandant": "military",
    "capitaine": "military",  # "less important" military positions
    "roi": "king",
    "empereur": "emperor",
    "president": "president",
    "homme politique": "politician",
    "président de l'assemblée": "politician",
    "orateur": "politician",
    "député": "politician",
    "secrétaire d'état": "politician",
    "sénateur": "politician",
    "écrivain": "writer",
    "auteur": "writer",
    "romancier": "writer",
    "acteur": "actor",
    "actrice": "actress",
    "cantatrice": "singer",
    "chanteur": "singer",
    "chanteuse": "singer",
    "peintre": "painter",
    "sculpteur": "sculptor",
    "statutaire": "sculptor",
    "compositeur": "composer",
    "musicien": "musician",
    "musicienne": "musician",
    "tragédien": "actor",
    "chansonnier": "chansonnier",
    "achitecte": "architect",
    "journaliste": "journalist",
    "inventeur": "inventor",
    "chimiste": "chemist",
    "connétable": "constable",
    "archevêque": "archbishop",
    "évêque": "bishop",
    "docteur": "physicist",
    "médecin": "physicist"
}
```

Code source B.1 – Table de conversion associant un métier à son équivalent normalisé

```
dpts = [
    "ain",
    "aisne",
    "allier",
    "basses-alpes",
    "hautes-alpes",
    "alpes-maritimes",
    "annepins",
    "provence",
    "ardèche",
    "ardennes",
    "arriège",
    "arno",
    "aube",
    "aude",
    "aveyron",
    "bouches-de-l'elbe",
    "bouches-de-l'escaut",
    "bouches-de-l'yssel",
    "bpuches-de-la-meuse",
    "bouches-du-rhin",
    "bouches-du-rhône",
    "bouches-du-weser",
    "calvados",
    "cantal",
    "charente",
    "charente-inférieure",
    "cher",
    "corrèze",
    "corse",
    "côte-d'or",
    "côtes-du-nord",
    "creuse",
    "deux-nèthes",
    "deux-sèvres",
    "doire",
    "dordogne",
    "doubs",
    "drôme",
    "dyle",
    "ems-occidental",
    "ems-oriental",
    "ems-supérieur",
    "escaut",
    "eure",
    "eure-et-loir",
    "finistère",
    "forêts",
    "gard",
    "haute-garonne",
    "gers",
    "gironde",
    "hérault",
```

```
countries = {
    "états-unis d'amérique": "united states of america",
    "etats-unis d'amérique": "united states of america",
    "états unis d'amérique": "united states of america",
    "etats unis d'amerique": "united states of america",
    "états-unis": "united states of america",
    "etats-unis": "united states of america",
    "etats unis": "united states of america",
    "états unis": "united states of america",
    "italie": "italy",
    "grèce": "greece",
    "canada": "canada",
    "chine": "china",
    "haïti": "haiti",
    "tobago": "tobago",
    "brésil": "brasil",
    "burkina-faso": "burkina-faso",
    "cameroun": "cameroun",
    "tchad": "tchad",
    "congo": "congo",
    "gabon": "gabon",
    "guinée": "guinea",
    "côte d'ivoire": "ivory coast",
    "mali": "mali",
    "mauritanie": "mauritania",
    "niger": "niger",
    "sénégal": "senegal",
    "madagascar": "madagascar",
    "seychelles": "seychelles",
    "tanzanie": "tanzania",
    "zanzibar": "zanzibar",
    "liban": "lebanon",
    "syrie": "syria",
    "inde": "india",
    "laos": "laos",
    "viet-nâm": "vietnam"
}
```

Code source B.3 – Table de conversion pour les pays

```
colonies = [
    "québec",
    "ontario",
    "saint-pierre-et-miquelon",
    "mississippi",
    "missouri",
    "louisiane",
    "anguilla",
    "antigua",
    "dominique",
    "saint-domingue",
    "guadeloupe",
    "monsterrat",
    "saint-martin",
    "saint-barthélémy",
    "sainte-lucy",
    "saint-vincent-et-les-grenadines",
    "saint-eustache",
    "saint-christophe",
    "martinique"
    "guyane française",
    "guyane",
    "maroc",  # unfortunately the morocco referred to in XIXth century
    ↪  france is a french protectorate
    "algérie",  # same
    "algérie française",  # same
    "tunisie",  # same
    "fezzan",
    "dahomey",
    "haute-volta",
    "oubangui-chari",
    "congo français",
    "moyen-congo",
    "guinée française",
    "soudan français",
    "gorée",
    "tigi",
    "djibouti",
    "cheikh saïd",
    "comores",
    "fort-dauphin",
    "îles maurice",
    "mayotte",
    "la réunion",
    "îles éparses",
    "île amsterdam",
    "île saint-paul",
    "archipel crozet",
    "îles kerguelen",
    "castellorizo",
    "grand-liban",
    "sandjak d'alexandrette",
    "indes françaises",
```

```
provinces = [
    "armagnac",
    "île-de-france",
    "berry",
    "orléanais",
    "normandie",
    "languedoc",
    "lyonnais",
    "dauphiné",
    "champagne",
    "aunis",
    "saintonge",
    "poitou",
    "guyenne et gascogne",
    "bourgogne",
    "picardie",
    "anjou",
    "provence",
    "angoumois",
    "bourbonnais",
    "marche",
    "bretagne",
    "maine",
    "touraine",
    "limousin",
    "comté de foix",
    "auvergne",
    "béarn",
    "alsace",
    "artois",
    "roussillon",
    "flandre française et hainaut français",
    "franche-comté",
    "lorraine et trois-évêchés",
    "corse",
    "nivernais",
]
```

Code source B.5 – Liste d'anciennes provinces françaises pour la détection de motifs

```
events = {
    "défense nationale": "government of national defense",
    "defense nationale": "government of national defense",
    "révolution française": "french revolution",
    "revolution francaise": "french revolution",
    "guerre de trente ans": "thirty years' war 1618 1648",
    "guerre de cent ans": "hundred years' war 1337 1453",
    "guerre de sept ans": "seven years war 1756 1763",
    "guerre": "war",
    "insurrection": "war",
    "siège de mayence": "siege of mainz",
    "siège": "siege",
    "commune": "commune",
    "défense": "battle",
    "révolution": "revolution"
}
```

Code source B.6 – Table de conversion pour les évènements historiques

```

```

Code source B.7 – Liste de colonies pour la détection de motifs

```

```

Code source B.8 – Liste de colonies pour la détection de motifs